An adaptive model switch-based surrogate-assisted evolutionary algorithm for noisy expensive multi-objective optimization

Nan Zheng · Handing Wang · Bo Yuan

Received: 22 August 2021 / Accepted: 27 February 2022 / Published online: 1 April 2022
© The Author(s) 2022

Abstract
To solve noisy and expensive multi-objective optimization problems, there are only a few function evaluations can be used due to the limitation of time and/or money. Because of the influence of noises, the evaluations are inaccurate. It is challenging for the existing surrogate-assisted evolutionary algorithms. Due to the influence of noises, the performance of the surrogate model constructed by these algorithms is degraded. At the same time, noises would mislead the evolution direction. More importantly, because of the limitations of function evaluations, noise treatment methods consuming many function evaluations cannot be applied. An adaptive model switch-based surrogate-assisted evolutionary algorithm is proposed to solve such problems in this paper. The algorithm establishes radial basis function networks for denoising. An adaptive model switch strategy is adopted to select suited surrogate model from Gaussian regression and radial basis function network. It adaptively selects the sampling strategies based on the maximum improvement in the convergence, diversity, and approximation uncertainty to make full use of the limited number of function evaluations. The experimental results on a set of test problems show that the proposed algorithm is more competitive than the five most advanced surrogate-assisted evolutionary algorithms.

Keywords Multi-objective optimization · Surrogate · Noise treatment · Evolutionary algorithms · Sampling approach

Introduction
Nowadays, there are many practical engineer problems that can be modeled as noisy and expensive multi-objective optimization problems (MOPs), such as the gait optimization of autonomous snake robot. It needs to maximize the speed and head stability of the sidewinding gait of a snake robot [1]. Because the snake robot achieves locomotion by twisting its body [2], its speed and head stability are a pair of conflicting objectives. The objective values can only be observed through experiments, so the cost of observations is expensive. Due to the influence of environment and robot gait, it is difficult to get accurate objective information from the speed and vision sensors. Therefore, this optimization problem can be modeled as an expensive MOP with independent noises on each objective.

MOPs usually have multiple conflicting objectives to be optimized at the same time. Evolutionary algorithms (EAs) [3] are widely used to solve MOPs because of some advantages. Those advantages include that obtaining a set of non-dominated solutions in one run [4], simultaneously handling problems with multiple local and non-convex Pareto fronts [5]. Therefore, a large number of multi-objective evolutionary algorithms (MOEAs) have proposed for MOPs in recent decades [6]. These MOEAs usually are divided into three types: algorithms based on Pareto domination, indictors and decomposition. The first type of MOEAs select candidate solutions by the Pareto dominance. For example, nondominated sorting genetic algorithm-II (NSGA-II) [7] uses the dominance relationship and crowding distance as criteria for its selection. The second type of MOEAs assign a fit-
SAEAs can be divided into two classes: According to the methods of constructing surrogates, these classification neural networks (ANN) [20], support vector machines (SVM) [18], Gaussian process (GP) [19], artificial neural networks (ANN) [20], support vector machines (SVM) [21], radial basis function networks (RBFN) [22]. The methods to construct surrogates include response surface method (RSM) [18], Gaussian process (GP) [19], artificial neural networks (ANN) [20], support vector machines (SVM) [21], radial basis function networks (RBFN) [22]. According to the methods of constructing surrogates, these SAEAs can be divided into two classes:

1. Regression-based SAEAs: These SAEAs use surrogates to approximate objective functions, indictors, or scalar functions, etc. The examples of the first category include Kriging-assisted two-archive evolutionary algorithm (KTA2) [23], Kriging-assisted reference vector guided evolutionary algorithm (KRVEA) [5]. KRVEA uses angle penalized distance (APD) [5] to select the solutions to keep the convergence and diversity of population. However, when the number of inactive fixed reference vectors, i.e., the reference vectors without any associated solution, the increasing number of those reference vectors between generations reaches a threshold, it would select the solutions with uncertainty. KTA2 considers convergence, diversity, and approximation uncertainty in its sampling criterion, which analyzes the optimization process need on convergence, diversity and approximation uncertainty then adaptively selects a sampling strategy based one aspect. ADSAPSO adopts the dominance relationship and crowding distance as the criteria sampling, which takes convergence and diversity into consideration. The examples of the second category include neural networks assisted steady state indicator-based evolutionary algorithm (NN-SS-IBEA) [24]. In the sampling criteria of NN-SS-IBEA, it choses individual with a high HV contribution which considers convergence and diversity. The examples of the third category include Pareto based efficient global optimization (ParEGO) [25], efficient global optimization-assisted multi-objective evolutionary algorithm based on decomposition (MOEA/D-EGO) [26]. ParEGO and MOEA/D-EGO select the solution which has the best expected improvement (EI) [27] of the aggregation function to sample, which takes both convergence and uncertainty into account.

2. Classification-based SAEAs: These SAEAs make use of the existing data to train a classifier. It is used as the surrogate model to select candidate solutions as their function evaluations, such as classification and Pareto-based MOEA (CPS-MOEA) [28], classification based on surrogate-assisted evolutionary algorithm (CSEA) [29]. CPS-MOEA selects the solutions with good fitness which mainly considers convergence. As there is no solution with good fitness, CPS-MOEA selects the solutions for function evaluations randomly which takes the approximation uncertainty into account. CSEA selects nondominated solutions with a high credibility level and dominated solutions with a low credibility level for function evaluations which takes convergence and approximation uncertainty into account. The main differences between CPS-MOEA and CSEA include their surrogates and sampling criteria. CPS-MOEA applies a classification tree as its surrogate but CSEA uses feedforward neural network (FNN) [29] as its surrogate. CPS-MOEA selects solutions with good fitness by surrogate but CSEA selects solutions by comprehensively considering the dominance relationship and credibility.

Noises would lead that the objective function cannot be accurately evaluated. Thus, the optimization direction may be misled [30]. To improve the performance of the MOEAs, we need to take some extra methods for dealing with noises for noisy MOPs. At present, the approaches for handling noises in MOEAs can be compartmentalized into three categories [31] as follow:

1. Averaging: The average value of all solutions after independently sampling several times is taken as the objective values after denoising. With this approach, resampling n times, then the standard deviation of the objectives would decrease $\sqrt{n}$ times [32]. This is a common method for
decreasing noises but it requires many function evaluations [33–35]. For example, noise-tolerant strength Pareto evolutionary algorithm (NTSPEA) [35], each solution is assigned resampling times according to the number of dominated solutions in the population.

2. Ranking: The ranking methods are used to decrease noises include two types: ranking based on probability [36–39] and clustering [40]. For the first one, constructing a model with confidence level which is applied to predict the dominated probability between solutions with noises. For example, nondominated sorting genetic algorithm—II-A (NSGA-II-A) [37] constructs an SVM model with a confidence level by ω dominance to predict the relationship between solutions with noises. The other one, the average and standard deviation of each solution are calculated by resampling, and the clustering radius is obtained by the calculation results, so as to select more reliable solutions. For instance, modified nondominated sorting genetic algorithm—II (M-NSGA-II) [40] uses this method to solve noisy MOPs.

3. Modeling: This approach considers that noises may have seriously influence on one solution but on a model can be minimal. Therefore, it uses the noisy data to construct a regression model for denoising. For example, regularity model [31], restrict Boltzmann machine [41], [42] and univariate marginal distribution algorithm [43]. They all handle noises through the idea of modeling which brings a new direction for denoising.

At present, there are some MOEAs that combine the above methods for denoising, such as Filters-based NSGA-II (FNSGA-II) [44]. It uses resampling, mean and Wiener filters for denoising, where resampling mean filter belong to averaging and Wiener filters belong to modeling. However, there are many MOEAs to solve either expensive MOPs or noisy MOPs. For noisy and expensive MOPs, these MOEAs cannot achieve satisfactory performance. According to the above analysis, we find that the MOEAs for expensive MOPs have poor performance for noisy MOPs due to the influence of noise, and the MOEAs for noisy MOPs which need a large number of evaluations cannot meet the computation budget limitation [31,33]. To solve this type of MOPs, we need to face three main challenges. First, because of the influence of noises, the reliability of surrogates would be reduced. Second, due to the limited number of function evaluations, some existing denoising methods are invalid. Third, most sampling criteria cannot make full use of limited function evaluations. Therefore, we propose an adaptive model switch-based surrogate-assisted evolutionary algorithm (MS-MOEA) in this paper. We adopt IBEA as the basic optimizer, where $L_{+}$ and $L_{p}$-norm distance [45] are adopted as indicators to select sampling solutions and the GP and RBFN models are used as the surrogates to approximate objective functions.

The main contributions of this work as follow:

- A noise treatment method is proposed for decreasing the influence of noises. The approach divides all solutions into two sets: one of which all solutions greatly affected by noises, the other contains all remaining solutions. When the number of solutions in the first set reaches a certain scale, then an RBFN model is constructed by all data, and the data is smoothened by the model for denoising.
- An adaptive model switch approach is designed for the proposed algorithm. Either the RBFN or GP model is chosen as the surrogate according to whether the data set is denoised or not.
- We design an improved environmental selection for the MS-MOEA. It combines different selection methods to select individuals according to the impact of noise on the population.
- We propose an adaptive sampling strategy in this paper. This strategy includes three sampling methods and an approach to adaptively switch between the sampling methods.

The rest of the paper is organized as follows. An introduction to noisy MOPs, IBEA, GP and RBFN is given in the next section. The details of the proposed algorithm for noisy and expensive MOPs are described in the following section. The experimental settings, results and analysis are presented in the next section. The conclusions and future work are concluded in the last section.

**Preliminaries**

**Noisy multi-objective optimization problems**

We consider the mathematical expression of MOPs as is follows:

\[
\begin{align*}
\text{Minimize} & \quad F(x) = (f_1(x), f_2(x), \ldots, f_m(x))^T, \\
\text{s.t.} & \quad x = (x_1, x_2, \ldots, x_d)^T, \\
\end{align*}
\]

where $x$ is decision vector and $d$ is the dimension of decision vector. $F(x)$ is the objective function which has $m$ optimization objectives ($m \geq 2$). Since the objectives usually are conflicting, many solutions cannot be compared by objective values. The concept of the Pareto dominance [46] is to compare the solutions of MOPs. For example, there are two different solutions $a$ and $b$, only $\forall i = 1, 2, \ldots, m$, $f_i(a) \leq f_i(b)$ and $F(a) \neq F(b)$, a Pareto dominates $b$.

The MOPs that we need to solve have noises in the objective space, so the expression of the optimization objectives...
is shown in Eq. (2).

\[ T(x) = F(x) + \beta, \]
\[ \beta \sim N(0, \sigma^2 I), \]  

(2)

where \( F(x) \) is our optimization objectives, but we can only observe \( T(x) \). In this paper, we assume the noises are independent additive noises. \( \beta \) meets the Gaussian distribution. \( I \) is an identity matrix.

**Indicator-based selection evolutionary algorithm**

The evolutionary search of expensive MOPs need the assistance of surrogate models, which means that the approximation error of the model would have a great impact on the search process. IBEA\(_{e+}\) uses \( I_{e+} \) as selection criterion rather than the Pareto dominance relationship. \( I_{e+} \) is the minimal difference that the dominance relationship between two solution sets become one weakly dominating the other:

\[
I_{e+}(A, B) = \min_{\epsilon+} \{ \forall x_2 \in B, \exists x_1 \in A : f_i(x_1) - \epsilon \leq f_i(x_2), \text{ for } i = 1, \ldots, m \},
\]  

(3)

where \( A \) and \( B \) are solution sets, respectively, and \( m \) is the number of objectives. \( I_{e+}(A, B) \) represents the minimum distance of each objective dimension when the dominance relationship between \( A \) and \( B \) is to be transformed into weak dominance.

Compared with the Pareto dominance, there are two advantages of \( I_{e+} \) as follows: it does not only represent the natural extension of the Pareto dominance but also transforms multiple objectives into one indicator which reduces the approximation errors of surrogate models to a certain extent. When the dominance relationship is adopted for ranking, it is necessary to ensure that the prediction objective error does not affect the size relation of each two individuals in every objective. However, \( I_{e+} \) can reduce the approximation errors of surrogate models to a certain extent and the reason as follows. If \( A \) (or \( B \)) only contains one solution \( x_1 \) (or \( x_2 \)), the Eq. (3) can be written as

\[
I(x_1, x_2) = \max(F(x_1) - F(x_2))
\]  

(4)

Thus, \( I_{e+} \) is adopted as the indicator, it is only necessary to ensure that the prediction objective error of the selected surrogate model does not affect the size relation of each two individuals in one objective. Therefore, the adaptive IBEA\(_{e+}\) is adopted as basic optimizer in this work. IBEA\(_{e+}\) uses \( I_{e+} \) as the indicator to assign the fitness for each individual in the population. The binary tournament selection is used to select the parents. It makes use of crossover and mutation to generate offsprings. The pseudo code of adaptive IBEA\(_{e+}\) is shown in Algorithm 1.

**Algorithm 1 Adaptive IBEA\(_{e+}\)**

**Require:**
1. Population size \( N \);
2. Fitness scaling factor \( k \);
3. Maximum number of generations \( g_{\text{max}} \);

**Ensure:**
4. Approximate Pareto set \( P_t \);
5. Initialization: Set generation counter \( t = 0 \). Randomly create an initial population \( P_t \) with size \( N \).
6. Calculate objective value: Calculate the objective values of each individual in \( P_t \) by objective function.
7. Assign fitness: Assign fitness \( F \) for each individual in \( P_t \) by Eq. (5).
8. while \( t \leq g_{\text{max}} \) do
9. Mating: Select individuals from \( P_t \) into the mating pool \( Q \) by binary tournament selection based on fitness.
10. Generate offspring: Generate the offspring \( O_t \) with size \( N \) by crossing and mutating the individuals in \( Q \).
11. Calculate objective value: Calculate the objective values of each individual in \( O_t \) by objective functions.
12. Archive: Archive \( P_t \) and \( O_t \) into \( A_t \).
13. Assign fitness: Assign fitness \( F \) for each individual in \( A_t \) by Eq. (5).
14. Environment selection:
15. while length \( (A_t) > N \) do
16. Delete individual: Delete the individual \( W \) with the minimal \( F \).
17. Update Fitness: Apply Eq. (6) to update the fitness for rest of individuals in \( A_t \).
18. end while
19. \( P_{t+1} = A_t \)
20. \( t = t + 1 \)
21. end while
22. \( P_t = P_t \)

According to the above analysis, it is proved that \( I_{e+} \) can be directly used in fitness calculation. IBEA\(_{e+}\) adopts Eq. (5) to assign fitness for individuals:

\[
F(x) = \sum_{x' \in P \setminus \{x\}} -e^{-I(x', x) / (k+c)},
\]

(5)

\[
c = \max_{x \in P} |I(x, x')|,
\]

where \( k \) represents a scaling factor \((k > 0)\), \( c \) is defined to improve the robustness of the algorithm. Due to the dominance maintaining of \( I_{e+} \). Therefore, the better solution would be allocated higher fitness, and them are easier to be selected in the population iteration. Because the size of population is fixed, we need to delete some individuals by the fitness. Therefore, when the composition of the population changes, the fitness of every individual need to be updated. The Eq. (6) is used to update the

\[
F(x) = F(x) + e^{-I(W, x) / (k+c)},
\]

(6)

\[
c = \max_{x \in P} |I(x, x')|,
\]
where $W$ is the individual that needs to be deleted, other parameters same like Eq. (5).

**Radial basis function network**

RBFN [47] is a type of artificial neural network which usually has three layers: an input layer, a hidden layer and an output layer. The structure of RBFN, as shown in Fig. 1. In the RBFN, each neuron of the hidden layer is a radial basis function that takes the distance from the input point to the corresponding center point as the input variable. The calculation formula of the RBFN output is shown in Eq. (7):

\[
\text{Output} = \sum_{i=1}^{N} w_i \phi_i + \text{Bias},
\]

\[
\phi_i(r) = \phi_i(|x - c|),
\]

where $N$ represents the number of neurons in hidden layer, $w_i$ is the weight of $i$th the hidden layer neuron and the output layer, $\phi_i$ represents the output of $i$th the hidden layer and its input $(|x - c|)$ is the distance of input point and $i$th center point. Bias is the bias of the whole network.

There are four types of kernel functions commonly used in RBFN: Gaussian, reflected sigmoid, multiquadric and inverse multiquadric kernels [48]. In this work, we use the most commonly used Gaussian kernel function as the radial basis function, as shown in Eq. (8):

\[
\phi(r) = e^{-\frac{r^2}{2\sigma^2}},
\]

where $r = |x - c|$, $\sigma$ is the spread rate used to control the smoothness of RBFN which decides the range of central point.

In this paper, generalized least square method [49] is used for calculating the weight of radial basis function network, as shown in Eq. (9). It is assumed that there are $K$ training points $X = [x^1, x^2, \ldots, x^K]$ and $Y = [y^1, y^2, \ldots, y^K]$. Because the decision space dimension of the MOPs is $d$ and the objective space dimension is $m$, it is necessary to construct $m$ RBFNs to replace all objective functions:

\[
W = (\Phi^T \Phi)^{-1} \Phi^T \cdot Y,
\]

\[
W = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}, \quad \Phi = \begin{bmatrix} \phi_1^1 & \phi_1^2 & \cdots & \phi_1^Q \\ \phi_2^1 & \phi_2^2 & \cdots & \phi_2^Q \\ \vdots & \vdots & \ddots & \vdots \\ \phi_n^1 & \phi_n^2 & \cdots & \phi_n^Q \end{bmatrix}, \quad Y = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^K \end{bmatrix},
\]

where $Q$ is the number of neuron in hidden layer, $K$ represents the number of training data, $\phi_j$ represents the output of the $j$th hidden layer neuron at sample $x^i$. $Q$ and the central point are determined by K-means clustering algorithm [50].

Taking the Gaussian kernel as an example, the farther the input vector from the center of RBF, the lower the activation degree of the node is, so the RBFN has the characteristics of local fitting [47]. In addition, the training speed of RBFN is fast and the amount of calculation is small, which is one of the advantages of using RBFN as a surrogate model.

**Gaussian process**

Gaussian process is a excellent performance modeling method. It cannot only provide the predicted value of the points needed to be predicted, but also provide the uncertainty of points. Therefore, GP has been widely used in SAEAs. In this paper, we adopt DACE [51] to construct the GP model.

It is assumed that there are $K$ training points $X = [x^1, x^2, \ldots, x^K]$ and $Y = [y^1, y^2, \ldots, y^K]$ in data set $S$. Because the decision space dimension of MOPs is $d$ and the objective space dimension is $m$, it is necessary to construct $m$ GPs to replace all objective functions. The Gaussian process model based on data set $S$ satisfies Eq. (10):

\[
y(x) = \mu(x) + \alpha(x),
\]

\[
\alpha(x) \sim N \left(0, \sigma^2 \right),
\]

where $\mu(x)$ is the predicted value of $x$ by the model, $\alpha(x)$ represents an independently Gaussian distribution with 0 as the mean value, $\sigma$ as the standard deviation.

All training samples need to be trained to construct GP. For any two training samples $x^i$ and $x^j$, the two random processes $\alpha(x^i)$ and $\alpha(x^j)$ are defined as Eq. (11):

\[
\text{cov} \left( \alpha \left(x^i\right), \alpha \left(x^j\right) \right) = \sigma^2 R \left( [R \left( x^i, x^j \right)] \right),
\]

where $R$ represents the correlation matrix consisting of the correlation function values of all samples. For any $x^i, x^j$, .
shown in Eq. (12) by optimizing the hyperparameters: 

$$R(x^i, x^j) = e^{-d(x^i, x^j)}, \ d(x^i, x^j) = \sum_{i=1}^{K} \left| x^i_k - x^j_k \right|^2.$$ 

All \( \theta_i \) represent the hyperparameters.

To improve the performance of the surrogate model on the data set \( S \), we need to maximize the likelihood function as shown in Eq. (13) by optimizing the hyperparameters \( \theta \):

$$\hat{\mu} = \frac{1}{2\sigma^2} (\mathbf{y}^T \mathbf{R}^{-1} \mathbf{y} - \mathbf{1}^T \mathbf{R}^{-1} \mathbf{1}),$$

$$\sigma^2 = \frac{1}{K} \left[ (\mathbf{y} - \mu)^T \mathbf{R}^{-1} (\mathbf{y} - \mu) \right].$$

Finally, the calculation formula of predicted values and variance is shown in Eq. (14):

$$\hat{y}(x) = \hat{\mu} + r^T(x) \mathbf{R}^{-1}(\mathbf{y} - \mu),$$

$$\hat{\sigma}^2 = \sigma^2 \left[ 1 - r^T(x) \mathbf{R}^{-1} r(x) + \frac{(1 - \mathbf{r}^T(x) \mathbf{R}^{-1} r(x))^2}{\mathbf{r}^T(x) \mathbf{R}^{-1} \mathbf{r}(x)} \right].$$

where \( r(x) = [R(x, x^1), R(x, x^2), \ldots, R(x, x^K)] \) represents the correlation matrix consisting of the forecast point and all sample points.

It is noteworthy that the computational complexity of training a GP is \( O(N^3) \) [52] and the algorithms for optimizing the hyperparameters \( \theta \) would increase it. Therefore, GP is not suitable for the high-dimensional problems.

**Proposed algorithm**

In this work, an adaptive model switch-based SAEA (MS-MOEA) is proposed for noisy expensive multi-objective optimization. In this SAEA, we adopt IBEA as the basic optimizer, an adaptive model switching strategy between RBFN and GP to approximate objective functions, an infill sampling criterion to select solutions for real function evaluations. The framework of the proposed MS-MOEA is shown in Fig. 2. From the figure, MS-MOEA can be divided into six main steps.

1. Initialization: An initial population \( P_0 \) in the size \( N \) is generated by Latin hypercube sampling [53]. All individuals in the \( P_0 \) have been evaluated by the real objective functions \( T(x) \) which are expensive and noisy, then the individuals of \( P_0 \) would be added to the archive \( A \).
2. Noise treatment: A solution with the best fitness is sampled several times, and the average value is calculated as the actual objective value of the solution each objective.
The absolute value of maximum noise $E_{\text{max}}$ can be calculated by the sampling value and mean value. To estimate the influence of noises, we need to calculate the ratio of $E_{\text{max}}$ to objective values and record it as $\lambda$ and then compare $\lambda$ and the threshold of noises $\delta$, when $\lambda > \delta$, a denoising method is applied for objective values. Conversely, it is unnecessary. The denoising method is to take the data with noises to construct RBFN and the regression value as the denoised objective value.

3. Model switch: According to the result of noises estimation to select the suitable surrogate model from GP and RBFN, where $\lambda > \delta$, the algorithm selects RBFN and sets model flag to 0, conversely, the algorithm selects GP and sets model flag to 1.

4. Model building/rebuilding: All original data and the model flag are used to construct a suitable regression model which is used to replace the real function evaluations for each objective.

5. Offspring generation: The offspring solutions $O_t$ can be produced by reproduction operators such as crossover and mutation. Then, the real function evaluations are replaced by the constructed models to predict the objective values of all individuals. Parental selection chooses individuals from $A$ based on $I_{t+}$ and newly sampled individuals, plus individuals in $A$ for randomly migrating to commonly form parental population. To ensure the convergence of population, we set the number of generations assisted by surrogates to 20. This parameter is set according to KRVEA [5].

6. Adaptive sampling: By analyzing the improvement of population convergence and diversity, then we adopt an adaptive sampling approach to switch in three strategies based on convergence, diversity, and uncertainty. The sampled solutions would be evaluated by real objective functions, then the results of evaluations would be collected into archive $A$.

7. Repeat Steps 2–6 until the number of function evaluations is met.

In the following sections, we will discuss the details of Preliminaries, Proposed Algorithm, Empirical studies, and Conclusions.

**Noise treatment**

As mentioned in “Noisy multi-objective optimization problems”, we need to handle the noises in MOPs. For the minimization MOPs, noises have slight influence to the population in the early stage of optimization, because the population has not converged and the objective values are relatively large. In addition, taking the view into consideration, this view holds that when the impact of noises is slight that is helpful to generate better solutions, but when it is too big that evolutionary search degenerates into random search [30]. Therefore, in order to reduce the influence of noises, we propose a method to estimate the impact of noises and a method to denoise.

According to the above analysis, a method is designed to estimate the influence of noise which divides the whole stage of optimization into early and late stages. When the algorithm is in the early stage of optimization, it means that the noise has little influence on the population, so the noise is not handled. Conversely, the algorithm would construct RBFNs with original data for all objectives and use the regression value as denoised objective value. The method of estimating the influence of the noise is to use the fitness of IBEA $\epsilon +$ to sort the individuals in the initial population and selects the individual with the best fitness to resample 5 times by real objective functions. Then, we calculate the average value and find out the absolute value of maximum noise $E_{\text{max}}$ by the average value and all sample values for each objective. Next, the $N$-th individual sorted is for the estimation. Then the algorithm calculates the ratio of $E_{\text{max}}$ to objective values and records it as $\lambda$ in each objective, where $N$ is the population size. Finally, we need to compare $\lambda$ with the threshold of noises $\delta$. As long as there is an objective $\lambda < \delta$, it means that the algorithm is in the early stage of optimization, and the algorithm sets the noise flag to 0. Conversely, the algorithm would set the noise flag to 1, and once the noise flag is 1, the noise estimation method is no longer executed.

The output of RBFN consists of the product of all hidden layer neurons and corresponding weights, plus network bias. The output of all hidden layer neurons is controlled by the distance between input point and hidden layer center point. It is beneficial for denoising, because the correlation of objective values are affected by the distance in the decision space and the noises only exists in the objective space. In addition, The effect of denoising method depends on the number of center points and the value of the spread rate.

Model switch strategy

For the minimization MOPs, the noises in the early stage of optimization has little influences. Therefore, we can ignore the influence of noises for evolutionary process in the early stage of optimization. GP is a surrogate model with excellent performance which cannot only provide the prediction, but also provide the prediction error as the approximation uncertainty. It is extremely important in the early stage of evolution especially when the population is trapped in the local optimum. However, when the noises have non-ignorable influence, GP cannot handle noises well due to the fitting of GP to the objective is overfitting. Thus, It is not appropriate to use GP as the surrogate model in the late stage of evolution. RBFN has the characteristics of local fitting can smooth the noisy data, so RBFN as the surrogate model in the late stage
of optimization can achieve better performance. Based on the above analysis, we propose a adaptive model switch strategy between GP and RBFN based on the influence of noises. If the noise flag is 0, it means that algorithm need to concentrate on the convergence of population and approximation uncertainty so the GP surrogate model would be constructed for MS-MOEA. If the noise flag is 1, it means that algorithm needs to concentrate on the influence of noises so the RBFN surrogate model would be constructed for MS-MOEA.

**Improved environmental selection**

In the offspring generation, we need to select next parent population by the environmental selection. It is very important to consider the convergence and diversity of population in environmental selection. However, the basic optimizer IBEA$e$ cannot balance both aspects which is not very effective in maintaining diversity [17]. Therefore, we proposed an improved environmental selection.

**Algorithm 2** Improved environmental selection

Require:  
1: Population size $N$;  
2: Noise flag $NF$;  
3: Parent population $P_p$;  
4: Offspring population $P_o$;  
Ensure:  
5: Parent population $P_g$;  
6: If $NF = 0$ then  
7: Evaluate: All individuals in populations are evaluated by GP.  
8: Environment selection  
9: Convergence selection: $N/2$ individuals are selected from $P_p$ and $P_o$ by $I_{c+}$.  
10: Uncertainty selection: $N/2$ individuals are selected from all the remaining individuals in $P_p$ and $P_o$ by the $MSE$ of all $m$ surrogates.  
11: Update $P_g$: Combine two groups of individuals as new $P_g$.  
12: else  
13: Evaluate: All individuals in populations are evaluated by RBFN.  
14: Environment selection  
15: Diversity selection: $N/2$ individuals are selected from $P_p$ and $P_o$ by dominance relationship and $L_p$-norm-based distance.  
16: Convergence selection: $N/2$ individuals are selected from all the remaining individuals in $P_p$ and $P_o$ by $I_{c-}$.  
17: Update $P_g$: Combine two groups of individuals as new $P_g$.  
18: end if  
19: Return $P_g$

It can be seen from the above analysis that the evolutionary stage of the population can be divided into the early and late stages according to the noise flag. When the population is in the early stage, the population has not converged, it is important for environmental selection to pay attention to convergence. At the same time, considering that the population may fall into the local optimum, environmental selection needs to take uncertainty into account. Therefore, we adopt environmental selection based on convergence and uncertainty, respectively. Where convergence selection depends on the $I_{c+}$ indicator and uncertainty selection depends on the prediction error of GP. All individuals selected by environmental selection are regarded as new parental populations. When the population is in the late stage of evolution, the population has converged well. To get better solutions, we need to improve the diversity of population. Thus, we adopt environmental selection based on convergence and diversity, respectively. Where diversity selection depends on dominance relationship and the $L_p$-norm-based distance [54]. All individuals selected by environmental selection are regarded as new parental populations. For details of the improved environment selection are shown in Algorithm 2.

**Adaptive sampling approach**

Different sampling approaches can bring different improvements to the population or surrogate model. For example, the convergence sampling and diversity sampling approaches are helpful to improve the performance of population in convergence and diversity, while the uncertainty sampling approach is helpful to improve the global approximation accuracy of surrogate models. Because the MOPs studied in this work are expensive, we need to make good use of limited function evaluations and different sampling approaches. Therefore, an adaptive sampling approach is proposed for MS-MOEA. The procedure of adaptive sampling as shown in Fig. 3.

The idea of adaptive sampling approach is through preserve the populations that obtained by the convergence selection and diversity selection last time in the adaptive sampling are recorded as $P^{s-1}_c$ and $P^{s-1}_d$. At the same time, two selection approaches are used to select populations which are regarded as $P^g$ and $P^g_d$. $P^{s-1}_c$ and $P^{s-1}_d$ are used to compare the improvement of convergence. If there is an improvement from $P^{s-1}_c$ to $P^g$, the convergence sampling would be adopted to sample. When there is not an improvement from $P^{s-1}_c$ to $P^g$, the algorithm would use the noise flag to judge the evolutionary stage of the population to make a choice from adopting the diversity sampling directly and comparing the improvement of diversity. If there is an improvement from $P^{s-1}_d$ to $P^g$, the diversity sampling would be adopted. When there is not an improvement from $P^{s-1}_d$ to $P^g$, the approach would choose uncertainty sampling.

As for the methods to judge the improvement of convergence and diversity, we adopt two indicators to measure. To measure the improvement of convergence, we calculate the distances from all solutions to the estimated ideal point. To measure the improvement of diversity, we calculate the pure diversity (PD) [55] for each population. The calculation formula of PD is shown in Eq. (15):
Fig. 3 Procedure of adaptive sampling

\[ PD(P) = \max_{s_i \in X} \left( PD(P - s_i) + d(s_i, P - s_i) \right) \]
\[ d(s, P) = \min_{n_i \in P} \left( \text{dis}(s, s_i) \right) \]

where \( \text{dis}(s, s_i) \) is the \( L_p \) norm between individual \( s \) and individual \( s_i \). \( d(s, P) \) is used to obtain the minimum \( L_p \) norm. \( PD \) is an indicator to measure the diversity of population which does not require any reference, nor any parameters.

For the judgement results, the algorithm uses a specific sampling approach to select \( \eta \) individuals for expensive function evaluations:

1. Convergence sampling approach deletes the individual with minimum fitness calculated by \( I(\epsilon + \text{in}) \) in \( P_{g-1}^c \) until \( \eta \) individuals in \( P_g^c \). The details can be found in Algorithm 3.

2. Diversity sampling approach selects the nondominated individual with maximum contribution to the PD of \( P_g^d \) until the number of the selected individuals equals to \( \eta \). The details can be found in Algorithm 4.

3. Uncertainty sampling approach randomly selects \( \chi \) solutions from \( P_d^g \). Next, the solution with the maximum uncertainty would be selected in \( \chi \) solutions. The approach would be repeated \( \eta \) times and \( \eta \) samples are used to evaluate. The details can be found in Algorithm 5.

**Empirical studies**

To verify the performance of the proposed algorithm, we compare it with five SAEAs. The five algorithms include KRVEA [5], ParEGO [25], MOEA/D-EGO [26], CSEA [29], and KTA2 [17]. The characteristics of these algorithms are summarized as follows:

- **KRVEA**: The reference vector guided evolutionary algorithm (RVEA) [56] is adopted as the basic optimizer. The
The inverted generational distance (IGD) is adopted for comprehensive indicator that can consider both convergence and diversity of the nondominated solutions. IGD is defined as an effective indicator to evaluate.

**Parameter settings**

To ensure the fairness of the experiments, the common parameter setting of all compared SAEAs are listed below:

1. The initial population size \( N \) and the number of initial data are set to 100.
2. The number of expensive function evaluations is set to 300.
3. The variance of additive noise is set to 0.2.

In addition, the other parameter settings of the five compared SAEAs are set according to the recommended settings in the original literature. The key parameters of the proposed algorithm are set as follows:

1. The fitness scaling factor of IBEA \( \epsilon \) is set to 0.05.
2. The number of individuals for sampling \( n \) is set to 5.
3. The hyperparameter range of Matlab Toolbox DACE for GP model is set to \([10^{-5}, 10^5]\). We adopt the algorithm raised by Hookes and Jeeves [58] to optimize the hyperparameters.
4. The number of center points of RBFN hidden layer is set to \( \sqrt{\text{size}(D)} \), \( D \) is the number of the training data.
5. The spread rate \( \sigma \) is set to 2*Dis, Dis is the maximum distance between all center points.
6. The number of model assisted iteration \( w_{\text{max}} \) is set to 20.
7. The threshold of noise \( \delta \) is set to 0.03.
8. The parameters of crossover and mutation are set to \( p_c = 1, \eta_c = 20, p_m = 1, \eta_m = 20 \).

MS-MOEA has a parameter \( \delta \) which directly affects the performance of noise treatment. It is used for estimating the influence of noise. Therefore, we set an experiment on DTLZ1 and DTLZ2 to select a proper value for the threshold of noise \( \delta \). DTLZ1 is a hard-to-converge test function. It is usually difficult to converge to the late stage of evolution when the noise has non-ignorable impact. DTLZ2 is an easy-to-converge test function. It is easy to converge to the late stage of evolution. That is the reason why we select them as the test functions. To better compare the effects of different noise thresholds on the performance of the proposed algorithm, we take a set of \( \delta = \{0.0003, 0.003, 0.03, 0.13, 0.23, 0.33, 0.43, 0.53\} \) for experiments. For the convenience of observation, we set the \( x \)-axis coordinate to \( \lg(x) \). Figure 4 is the result of DTLZ1 and DTLZ2 with two objectives which is executed by 30 runs independently.

From Fig. 4, we can discover that \( \delta \) determines the sensitivity of noise estimation method. The smaller this parameter \( \delta \) is set, the more sensitive it is to noises. From Fig. 4a, we can discover this conclusion. When \( \delta \geq 0.03 \), it gets the familiar results in DTLZ1. When \( \delta < 0.03 \), the algorithm performance
is significantly reduced. The larger this parameter $\delta$ is set, the more non-sensitive it is to noises. From Fig. 4b, we can discover the conclusion. When $\delta = 0.03$, it gets the best results in DTLZ2. When $\delta$ is set be larger, the noise needs to cause greater impact on the population before denoising. For keeping the suitable sensitivity of noise estimation method, $\delta$ is set to 0.03.

**Effect of noise treatment**

To test the performance of the proposed noise treatment, we design an experiment that selects DTLZ2 and DTLZ5 as the test problems. Because DTLZ2 and DTLZ5 belong to the test problems that are sensitive to noises. Every test problem would be tested 30 runs independently. MS-MOEA and the version without noise treatment, termed N-(MS-MOEA), are compared. To ensure the fairness of the experiment, the objective values without noises are used in the calculation of IGD, and the real objective values do not participate in population iteration. The results are shown in Table 1.

The experimental results are as expected, MS-MOEA have achieved the best results on all test problems. It proves the effectiveness of our proposed noise treatment method. Noises in the objective space have a direct impact on the dominance relationship between individuals in the population. It would has an adverse impact on the evolutionary search of the population, and even degenerate into random search. Therefore, when the noises has a large influence on the objective values, it is very necessary to deal with the noises.

**Effect of model switch strategy**

To discuss the role of model switch in MS-MOEA, we set up the following experiment. This experiment adopts DTLZ1, DTLZ2 and DTLZ5 as the test problems. Each problem would be tested 30 runs independently. According to the different models used in the experiment, the experimental variables are recorded as (MS-MOEA)-GP, (MS-MOEA)-RBFN and MS-MOEA. To ensure the fairness of the experiment, the objective values without noises are used in the calculation of IGD, and the objective values do not participate in the optimization process. In addition, the frequency of usage of the two surrogate models in MS-MOEA is used to further explain the experimental results. The meaning of FG and FR is shown below. The results are shown in Table 2.

- FG: It is the frequency of the usage of GP model.
- FR: It is the frequency of the usage of RBFN model.

According to the results of experiment, we discover that the model switch strategy can find better surrogate model between GP and RBFN. It also proves that selecting an appropriate model according to the influence of noise on the objective values is conducive to evolutionary optimization. In addition, we can find that the performance of MS-MOEA and (MS-MOEA)-GP is similar in DTLZ1, it is because DTLZ1 is a hard-converging test function. Therefore, it is always in the early stage of the evolution process, when the impact of the noise is little. That is why the RBFN is never used in DTLZ1. In addition, we can discover the performance of using GP as the surrogate model is better than using RBFN as the surrogate model from the experimental results. The performance of MS-MOEA and (MS-MOEA)-RBFN is similar in DTLZ2 and DTLZ5, because both of them are sensitive to the noise. Therefore, they are always in the late stage of the evolution process, when the impact of the noise is non-ignorable. That

![Fig. 4](image-url)

Fig. 4 a is the performance profile plots of DTLZ1. b is the performance profile plots of DTLZ2 Average IGD profile plots over the different parameter $\delta$ in MS-MOEA on the two objectives DTLZ1 and DTLZ2 based on 30 independent runs.

---

**Table 1 Results of N-(MS-MOEA) and MS-MOEA on the DTLZ2 and DTLZ5 problems**

| Problems  | M    | N-(MS-MOEA)                          | MS-MOEA        |
|-----------|------|--------------------------------------|----------------|
| DTLZ2     | 2    | 2.11E−01 (6.26E−02)+                 | 1.00E−01 (3.25E−02) |
|           | 3    | 3.27E−01 (6.95E−02)+                 | 1.74E−01 (3.57E−02) |
| DTLZ5     | 2    | 2.02E−01 (5.90E−02)+                 | 9.49E−02 (2.23E−02) |
|           | 3    | 2.22E−01 (6.54E−02)+                 | 1.22E−01 (4.90E−02) |

$+/≈/−$

The best result in each problem is bold. The symbols “+”, “−” and “≈” indicate that MS-MOEA is statistically significantly superior to, inferior to, and almost equivalent to N-(MS-MOEA), respectively (the significance level is 0.05).
is why the GP is never used in DTLZ2 and DTLZ5. In addition, we can discover the performance of using RBFN as the surrogate model is better than using GP as the surrogate model from the experimental results. The results can prove that our model switch strategy can skillfully use the influence of noise to select the appropriate surrogate model for the evolutionary optimization.

Effect of improved environment selection

To verify the role of the improved environment selection, we design an experiment. The conventional environment selection methods used in different evolutionary stages in the algorithm are compared. This experiment adopts DTLZ1, DTLZ2, DTLZ5 and DTLZ7 as the test problems. Each problem would be tested 30 runs independently, too. To ensure the fairness of the experiment, the objective values without noises are used in the calculation of IGD, and the real objective values does not participate in all the compared algorithms. The results are shown in Table 3.

- (MS-MOEA)-C+C: It only adopts convergence selection in all evolutionary stages.
- (MS-MOEA)-C+D: It adopts convergence selection in the early stage and diversity selection in the late stage.
- (MS-MOEA)-U+C: It adopts uncertainty selection in the early stage and convergence selection in the late stage.
- (MS-MOEA)-U+D: It adopts uncertainty selection in the early stage and diversity selection in the late stage.

As expected, MS-MOEA almost makes the best results in all the variants. From the results of experiments, (MS-MOEA)-C+C and MS-MOEA have similar performance on DTLZ1, because DTLZ1 is a difficult test problem, it is better to adopt a convergence selection in all evolutionary stages. We can also find that (MS-MOEA)-C+C and (MS-MOEA)-U+C have similar performance in DTLZ2 and DTLZ5. This is mainly because DTLZ2 and DTLZ5 are sensitive to noises. Therefore, the environmental selection in the late stage plays an important role. As for the reason why MS-MOEA is not significantly better than other algorithms on DTLZ2 and DTLZ5 is that the noise treatment can only deal with noises to a certain extent. Some objectives of DTLZ7 are sensitive to noise and some objectives are insensitive. Our improved environment selection uses different environmental selections for different evolutionary stages. Therefore, it has achieved satisfactory results. In summary, the selection considering convergence, diversity and uncertainty are the key to achieve the most satisfactory performance of MS-MOEA.
Table 3 Results of (MS-MOEA)-C+C, (MS-MOEA)-C+D, (MS-MOEA)-U+C, (MS-MOEA)-U+D and MS-MOEA on the DTLZ2 and DTLZ5 problems

| Problems | M | (MS-MOEA)-C+C | (MS-MOEA)-C+D | (MS-MOEA)-U+C | (MS-MOEA)-U+D | MS-MOEA |
|----------|---|-------------|-------------|-------------|-------------|--------|
| DTLZ1    | 2 | 2.43E+01 (1.09E+01) | 1.78E-01 (6.70E-02) | 1.00E+01 (3.25E-02) | 9.49E-02 (2.23E-02) | 1.22E+00 (1.83E+00) |
|          | 3 | 1.77E+01 (9.63E+00) | 1.34E-01 (9.47E-02) | 1.33E+00 (1.45E+00) | 1.22E+00 (1.45E+00) |        |
| DTLZ2    | 2 | 1.09E-01 (4.36E-02) | 1.09E-01 (4.82E-02) | 1.09E-01 (3.25E-02) | 1.09E-01 (3.25E-02) |        |
|          | 3 | 1.77E-01 (4.77E-02) | 2.35E-01 (6.00E-02) | 1.93E-01 (5.82E-02) | 1.74E-01 (3.57E-02) |        |
| DTLZ5    | 2 | 9.60E-02 (3.51E-02) | 1.15E-01 (2.57E-02) | 9.63E-01 (2.57E-02) | 1.24E-01 (9.47E-02) |        |
|          | 3 | 1.23E-01 (3.17E-02) | 1.45E-01 (5.31E-02) | 1.22E-01 (3.42E-02) | 1.22E-01 (3.42E-02) |        |
| DTLZ7    | 2 | 2.36E+00 (1.97E+00) | 1.53E+00 (1.42E+00) | 1.53E+00 (1.42E+00) | 1.53E+00 (1.42E+00) |        |
|          | 3 | 1.64E+00 (1.97E+00) | 1.53E+00 (1.42E+00) | 1.53E+00 (1.42E+00) | 1.53E+00 (1.42E+00) |        |

The best result in each problem is bold. The symbols “+,” “−” and “≈” indicate that MS-MOEA is statistically significantly superior to, inferior to, and almost equivalent to the others, respectively (the significance level is 0.05).

Effect of adaptive sampling approach

To test the performance of the adaptive sampling approach in MS-MOEA, MS-MOEA is compared with MS-MOEA without adaptive sampling approach that is regarded as (MS-MOEA)-N. This experiment adopts DTLZ1, DTLZ2 and DTLZ7 as the test problems. Each problem would be tested 30 runs independently. To ensure the fairness of the experiment, the objective value without noises is used in the calculation of IGD, and the objective value does not participate in the optimization process. The results are shown in Table 4.

By analyzing the experimental results, we discover that the algorithm with adaptive sampling approach have better performance than the other one. Because the adaptive sampling can find the improvement of convergence and diversity and use targeted sampling approach to select solutions to evaluate. It is beneficial to make full use of the limited function evaluations. MS-MOEA get the best results on all test problems.

To study the usage of different sampling approach in every test function, we summarize the results of 30 independent runs about them and these results are displayed in Fig. 5. It is easy to be discovered that the convergence sampling approach is most frequently used in DTLZ1, the uncertainty sampling approach is next and the diversity sampling approach is the last. DTLZ1 is a multi-modal test function, so it is difficult to converge and easy to fall into a local optimum. In DTLZ2, the diversity sampling approach is also more adopted. Because DTLZ2 is an easy to converge test function, it needs to take both convergence and diversity into account. The most frequently adopted sampling approach in DTLZ7 is the diversity sampling approach. The real PF of DTLZ7 is irregular. Therefore, the algorithm needs to pay attention to the maintenance of diversity. According to Fig. 5 and Table 4, our adaptive sampling approach can select appropriate sampling approach from the convergence approach, the diversity approach and the uncertainty approach based on the characteristic of the test functions.

Comparative experiments

To test the overall performance of the proposed algorithm, we employ KRVEA, ParEGO, MOEA/D-EGO, CSEA and KTA2 as compared algorithms and carry out experiments on the test problems DTLZ1∼7. This paper uses the compared algorithms implemented in PlatEMO [59] to test. The number of objectives includes two and three in this experiment. IGD is used as performance indicator for 14 test problems, and each problem would be tested 30 runs independently. For ensuring the fairness of the experiment, the objective values without noises is used in the calculation of IGD, and the
Table 4 Results of (MS-MOEA)-N and MS-MOEA on the DTLZ1, DTLZ2 and DTLZ7 problems

| Problems | M | (MS-MOEA)-N | MS-MOEA |
|----------|---|-------------|---------|
| DTLZ1    | 2 | 4.22E+01 (1.87E+01)+ | 2.43E+01 (1.09E+01) |
| DTLZ2    | 2 | 1.15E−01 (3.36E−02)+ | 8.45E−02 (3.31E−02) |
| DTLZ7    | 2 | 2.05E−01 (2.75E−01)≈ | 1.73E−01 (2.45E−01) |

The best result in each problem is bold. The symbols “+”, “−” and “≈” indicate that MS-MOEA is statistically significantly superior to, inferior to, and almost equivalent to the others, respectively (the significance level is 0.05).

From the experimental results, we can observe that MS-MOEA obtains the best results of 12 test problems; ParEGO obtains the best results of 1 test problems; KTA2 obtains the best result on 1 test problems; KRVEA, MOEA/D-EGO and CSEA do not get the best results on any test problem. Therefore, MS-MOEA has achieved the best performance between all compared algorithms on the DTLZ problems. The final non-dominated solutions obtained by six algorithms on two objectives DTLZ2 in the run associated with the median IGD values are plotted in Fig. 6. In addition, Fig. 7 shows the final non-dominated solutions obtained by six algorithms on three objectives DTLZ1 in the run associated with the median IGD values.

According to the characteristics of DTLZ series test problems, all the above test problems can be divided into three categories:

1. DTLZ1 and DTLZ3 are the first category. This kind of test problems are multimodal and have a large number of local optimal solutions, so it is difficult to converge to the true PF. DTLZ3 is more difficult to converge than DTLZ1.

2. DTLZ2 and DTLZ4 are the second type of test problems, which are uni-modal and easy to converge, but it is difficult to maintain the diversity of the population, and DTLZ4 is sensitive to the initial data.

3. DTLZ5~7 are the third type. The PF of these test problems are irregular. The real PF of DTLZ5 and DTLZ6 are degenerated curves, while the real PF of DTLZ7 is discontinuous. These test problems require that the algorithm can better maintain the diversity of population.

For the first type of test problems, MS-MOEA almost achieves the best result. Through the analysis of the results, it is found that the reason for this result may be that the model switch strategy can select a suited surrogate model from GP and RBFN. At the same time, the adaptive sampling approach can make use of approximation uncertainty to get rid of the local optimum. In addition, KTA2 has the similar results with MS-MOEA when the number of objectives is three. Because both of DTLZ1 and DTLZ3 are difficult to converge, noise have little influence on them. However, when the number of objectives is two, MS-MOEA is significantly better than KTA2. Because KTA2 is designed for many-objectives problems.

For the second type of test problems, MS-MOEA achieves the best results which show that the proposed noise treatment is very effective. The adaptive sampling approach can also find the most improved aspect for sampling by analyzing the population of candidate solutions. However, other compared algorithms are greatly affected by noise, because they do not deal with the noise, resulting in the degradation of algorithms performance.

For the third type of test problems, MS-MOEA obtains the best results on DTLZ5 and DTLZ6, which benefits from the noise treatment, the improved environmental selection and adaptive sampling approach. However, the performance of DTLZ7 is inferior to ParEGO. Through the analysis of the experimental results, it is found that the reason is that the noise treatment designed in this paper is lack when some objectives are easy to converge and other objectives are dif-
Table 5 Results of KRVEA, ParEGO, MOEA/D-EGO, CSEA, KTA2 AND MS-MOEA on the DTLZ1~7 problems

| Problems | M   | KRVEA            | ParEGO            | MOEA/D-EGO        | CSEA            | KTA2            | MS-MOEA          |
|----------|-----|-----------------|-------------------|-------------------|-----------------|-----------------|------------------|
| DTLZ1    | 2   | 3.28E+01 (1.49E+01)+ | 5.67E+01 (5.89E+00)+ | 6.10E+01 (1.04E+01)+ | 4.28E+01 (1.23E+01)+ | 3.63E+01 (1.15E+01)+ | 2.43E+01 (1.09E+01) |
|          | 3   | 2.96E+01 (8.68E+00)+ | 3.80E+01 (5.71E+00)+ | 3.94E+01 (9.61E+00)+ | 2.52E+01 (8.83E+00)+ | 1.62E+01 (6.18E+00)≈ | 1.78E+01 (6.70E+00)  |
| DTLZ2    | 2   | 5.83E−01 (9.24E−02)+ | 6.75E−01 (1.33E−01)+ | 6.11E−01 (1.27E−01)+ | 5.97E−01 (1.06E−01)+ | 6.44E−01 (1.10E+00)+ | 1.00E−01 (3.25E−02)  |
|          | 3   | 6.12E−01 (8.38E−02)+ | 5.94E−01 (5.78E−02)+ | 5.57E−01 (8.20E−02)+ | 5.45E−01 (7.48E−02)+ | 6.43E−01 (6.92E−02)+ | 1.74E−01 (3.57E−02)  |
| DTLZ3    | 2   | 8.53E+01 (3.65E+01)+ | 1.24E+02 (2.17E+01)+ | 1.34E+02 (2.95E+01)+ | 1.05E+02 (2.95E+01)+ | 7.37E+01 (2.72E+01)+ | 5.26E+01 (2.30E+01)  |
|          | 3   | 7.51E+01 (2.29E+01)+ | 9.83E+01 (9.39E+00)+ | 1.01E+02 (2.16E+01)+ | 6.23E+01 (2.66E+01)+ | 4.34E+01 (1.92E+01)≈ | 4.09E+01 (2.16E+01)  |
| DTLZ4    | 2   | 1.01E+00 (1.98E−01)+ | 9.97E−01 (1.85E−01)+ | 9.06E−01 (1.52E−01)+ | 8.27E−01 (2.08E−01)≈ | 9.99E−01 (1.96E−01)+ | 7.71E−01 (2.32E−02)  |
|          | 3   | 9.20E−01 (1.86E−01)≈ | 1.09E+00 (1.17E−01)+ | 9.14E−01 (1.08E−01)≈ | 9.92E−01 (1.58E−01)≈ | 1.08E+00 (1.51E−01)+ | 8.87E−01 (1.11E−01)  |
| DTLZ5    | 2   | 5.82E−01 (1.02E−01)+ | 6.24E−01 (1.14E−01)+ | 5.74E−01 (1.06E−01)+ | 5.76E−01 (9.90E−02)+ | 6.33E−01 (1.14E−01)+ | 9.49E−02 (2.23E−02)  |
|          | 3   | 5.08E−01 (9.58E−02)+ | 5.01E+01 (4.71E−02)+ | 4.71E+01 (8.68E−02)+ | 4.48E+01 (7.99E−02)+ | 5.18E−01 (8.18E−02)+ | 1.22E−01 (4.90E−02)  |
| DTLZ6    | 2   | 5.78E+00 (7.75E−01)+ | 1.91E+00 (5.64E−01)+ | 3.87E+00 (9.45E−01)+ | 7.38E+00 (6.40E−01)+ | 5.36E+00 (1.48E+00)+ | 1.59E+00 (6.00E−01)  |
|          | 3   | 5.52E+00 (6.45E−01)+ | 1.93E+00 (4.27E−01)+ | 3.76E+00 (8.02E−01)+ | 7.54E+00 (5.09E−01)+ | 4.77E+00 (8.74E−01)+ | 1.77E+00 (4.54E−01)  |
| DTLZ7    | 2   | 4.55E+00 (6.69E−01)+ | 3.26E−01 (9.63E−02)+ | 3.95E+00 (1.26E+00)+ | 2.56E+00 (7.04E−01)+ | 4.59E+00 (1.09E+00)+ | 1.33E−01 (1.82E−01)  |
|          | 3   | 5.52E+00 (1.71E+00)+ | 4.17E−01 (8.32E−02)+ | 4.91E+00 (1.80E+00)+ | 3.57E+00 (1.14E+00)+ | 6.24E+00 (1.45E+00)+ | 1.22E+00 (1.45E+00)  |

The best result in each problem is bold. The symbols “+”, “−” and “≈” indicate that MS-MOEA is statistically significantly superior to, inferior to, and almost equivalent to the others, respectively (the significance level is 0.05)
Fig. 6  Non-dominated front obtained by each algorithm on DTLZ2 with two objectives in the run associated with the median IGD value, respectively

Fig. 7  Non-dominated front obtained by each algorithm on DTLZ1 with three objectives in the run associated with the median IGD value, respectively
difficult to converge in the face of the test problem, which may be the reason why MS-MOEA is inferior to ParEGO.

Conclusions

In this work, we propose an adaptive model switch-based surrogate-assisted evolutionary algorithm called MS-MOEA for noisy expensive multi-objective optimization problems. We propose a method to estimate the influence of noise and a noise reduction method based on RBFN, and the noise experiment shows that this noise treatment is effective. This algorithm adopts a model switch strategy based on noise influence to select the surrogate model that meets the needs from GP and RBFN. We find that this strategy is helpful to improve the performance of the algorithm by the experiment. The proposed improved environmental selection and adaptive sampling approach are used to obtain candidates and samples. Through the comparison between MS-MOEA and the other five SAEAs, it is found that MS-MOEA achieves better performance than the compared SAEAs on noisy DTLZ problems.

However, there are still two deficiencies that can be improved in this paper. We need to improve the adaptive sampling approach for multimodal MOPs, these problems require the algorithm to use approximation uncertainty to search better solutions. For the same MOP, there are both noise sensitive and noise insensitive objectives, the noise treatment method performs poorly. In addition, the noise studied in this paper belongs to additive noises. Compared with the additive noise, the multiplicative noise is a more challenging task to be addressed.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Arizumi R, Tesch M, Kato K, Choset H, Matsuno F (2016) Multiobjective optimization based on expensive robotic experiments under heteroscedastic noise. IEEE Trans Rob 33(2):468–483
2. Bing Z, Cheng L, Huang K, Jiang Z, Chen G, Rührbein F, Knoll A (2017) Towards autonomous locomotion: Slithering gait design of a snake-like robot for target observation and tracking. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 2698–2703
3. Coello CAC, Lamont GB, Van Veldhuizen DA et al (2017) Evolutionary algorithms for solving multi-objective problems, vol 5. Springer, New York
4. Yang S, Li M, Liu X, Zheng J (2013) A grid-based evolutionary algorithm for many-objective optimization. IEEE Trans Evol Comput 17(5):721–736
5. Chugh T, Jin Y, Miettinen K, Hakanen J, Sindhya K (2016) A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. IEEE Trans Evol Comput 22(1):129–142
6. Coello Coello CA, González Brambila S, Figueroa Gamboa J, Castillo Tapia MG, Hernández Gómez R (2020) Evolutionary multiobjective optimization: open research areas and some challenges lying ahead. Complex Intell Syst 6(2):221–236
7. Deb K, Pratap A, Agarwal S, Meyarivan TAMT (2002) A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Trans Evol Comput 6(2):182–197
8. Zitzler E, Künzli S (2004) Indicator-based selection in multiobjective search. In: International conference on parallel problem solving from nature, pp 832–842. Springer
9. Bader J, Zitzler E (2011) Hype: an algorithm for fast hypervolume-based many-objective optimization. Evol Comput 19(1):45–76
10. Zitzler E, Thiele L, Laumanns M, Fonseca CM, Da Fonseca VG (2003) Performance assessment of multiobjective optimizers: an analysis and review. IEEE Trans Evol Comput 7(2):117–132
11. Zhang Q, Li H (2007) Moea/d: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evol Comput 11(6):712–731
12. Wang H, He S, Yao X (2017) Nadir point estimation for many-objective optimization problems based on emphasized critical regions. Soft Comput 21(9):2283–2295
13. Jin Y, Wang H, Chugh T, Guo D, Miettinen K (2018) Data-driven evolutionary optimization: an overview and case studies. IEEE Trans Evol Comput 23(3):442–458
14. Grefenstette JJ, Michael FJ (1985) Genetic search with approximate function evaluations. In: Proceedings of an international conference on genetic algorithms and their applications, vol 112, p 120
15. Knowles J, Nakayama H (2008) Meta-modeling in multiobjective optimization. In: Multiobjective optimization. Springer, pp 245–284
16. Jin Y, Olhofer M, Sendhoff B (2000) On evolutionary optimization with approximate fitness functions. In: GECCO, pp 786–793
17. Song Z, Wang H, He C, Jin Y (2021) A kriging-assisted two-archive evolutionary algorithm for expensive many-objective optimization. IEEE Trans Evolut Comput
18. Box GEP, Richard DN et al (1987) Empirical model-building and response surfaces, vol 424. Wiley, New York
19. Krige DG (1951) A statistical approach to some basic mine valuation problems on the witwatersrand. J S Afr Inst Min Metall 52(6):119–139
20. Zurada JM (1992) Introduction to artificial neural systems, vol 8. West, St Paul
21. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297
22. Broomhead DS, Lowe D (1988) Radial basis functions, multivariable functional interpolation and adaptive networks. Technical report, Royal Signals and Radar Establishment Malvern (United Kingdom)
23. Lin J, He C, Cheng R (2021) Adaptive dropout for high-dimensional expensive multiobjective optimization. Complex Intell Syst 1–15
24. Azzouz N, Bechikh S, Ben SL (2014) Steady state ibea assisted by mlp neural networks for expensive multi-objective optimization
25. Knowles J (2006) Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. IEEE Trans Evol Comput 10(1):50–66
26. Zhang Q, Liu W, Tsang E, Virginas B (2009) Expensive multiobjective optimization by moea/d with gaussian process model. IEEE Trans Evol Comput 14(3):456–474
27. Močkus J (1975) On bayesian methods for seeking the extremum. In: Optimization techniques IFIP technical conference. Springer, pp 400–404
28. Zhang J, Zhou A, Zhang G (2015) A classification and pareto dominance based multiobjective evolutionary algorithm. In: 2015 IEEE congress on evolutionary computation (CEC). IEEE, pp 2883–2890
29. Pan L, He C, Tian Y, Wang H, Zhang X, Jin Y (2018) A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization. IEEE Trans Evol Comput 23(1):74–88
30. Goh CK, Tan KC (2007) An investigation on noisy environments in evolutionary multiobjective optimization. IEEE Trans Evol Comput 11(3):354–381
31. Wang H, Zhang Q, Jiao L, Yao X (2015) Regularity model for noisy multiobjective optimization. IEEE Trans Cybern 46(9):1997–2009
32. Jin Y, Branke J (2005) Evolutionary optimization in uncertain environments-a survey. IEEE Trans Evol Comput 9(3):303–317
33. Syberfeldt A, Ng A, John RI, Moore P (2010) Evolutionary optimisation of noisy multi-objective problems using confidence-based dynamic resampling. Eur J Oper Res 204(3):533–544
34. Singh A (2003) Uncertainty based multi-objective optimization of groundwater remediation design. PhD thesis, University of Illinois at Urbana-Champaign
35. Buche D, Stoll P, Dornberger R (2002) Multiobjective evolutionary algorithm for the optimization of noisy combustion processes. IEEE Trans Syst Man Cybern Part C (Applications and Reviews) 32(4):460–473
36. Hughes EJ (2001) Evolutionary multi-objective ranking with uncertainty and noise. In: International conference on evolutionary multi-criterion optimization. Springer, pp 329–343
37. Boonma P, Suzuki J (2009) A confidence-based dominance operator in evolutionary algorithms for noisy multiobjective optimization problems. In: 2009 21st IEEE international conference on tools with artificial intelligence. IEEE, pp 387–394
38. Eskandari H, Geiger CD (2009) Evolutionary multiobjective optimization in noisy problem environments. J Heuristics 15(6):559
39. Basseur M, Zitzler E (2006) A preliminary study on handling uncertainty in indicator-based multiobjective optimization. In: Workshops on applications of evolutionary computation. Springer, pp 727–739
40. Babbar M, Lakshminantha A, Goldberg DE (2003) A modified nsga-ii to solve noisy multiobjective problems. In: 2003 genetic and evolutionary computation conference. Late-Breaking Papers, pp 21–27
41. Tang H, Ann SV, Chen TK, Yong CJ (2010) Restricted Boltzmann machine based algorithm for multi-objective optimization. In: IEEE congress on evolutionary computation. IEEE, pp 1–8
42. Shim VA, Tan KC, Chia JY, Al Mamun A (2013) Multi-objective optimization with estimation of distribution algorithm in a noisy environment. Evol Comput 21(1):149–177
43. Hong Y, Ren Q, Zeng J (2005) Optimization of noisy fitness functions with univariate marginal distribution algorithm. In: 2005 IEEE congress on evolutionary computation, vol 2. IEEE, pp 1410–1417
44. Liu R, Li Y, Wang H, Liu J (2021) A noisy multi-objective optimization algorithm based on mean and wiener filters. Knowl Based Syst 107215
45. Aggarwal CC, Hinneburg A, Keim DA (2001) On the surprising behavior of distance metrics in high dimensional space. In: International conference on database theory. Springer, pp 420–434
46. Le Yu P (1974) Cone convexity, cone extreme points, and non-dominated solutions in decision problems with multiobjectives. J Optim Theory Appl 14(3):319–377
47. Bishop CM et al (1995) Neural networks for pattern recognition. Oxford University Press, Oxford
48. Huang P, Wang H, Ma W (2019) Stochastic ranking for offline data-driven evolutionary optimization using radial basis function networks with multiple kernels. In: 2019 IEEE symposium series on computational intelligence (SSCI). IEEE, pp 2050–2057
49. Du K-L, Swamy MNS (2006) Neural networks in a softcomputing framework. Springer Science & Business Media, London
50. Shen W, Guo X, Chao W, Desheng W (2011) Forecasting stock indices using radial basis function neural networks optimized by artificial fish swarm algorithm. Knowl Based Syst 24(3):378–385
51. Lophaven N, Nielsen HB (2002) Sondergaard, “dace, a matlab kriging toolbox”. Technical University of Denmark, Technical Report, IM-REP-2002–12
52. Hensman J, Fusi N, Lawrence ND (2013) Gaussian processes for big data. arXiv:1309.6835
53. McKay MD, Beckman RJ, Conover WJ (2000) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 42(1):55–61
54. Wang H, Jiao L, Yao X (2014) Two.arch2: an improved two-archive algorithm for many-objective optimization. IEEE Trans Evol Comput 19(4):524–541
55. Wang H, Jin Y, Yao X (2016) Diversity assessment in manyobjective optimization. IEEE Trans Cybern 47(6):1510–1522
56. Cheng R, Jin Y, Olhofer M, Sendhoff B (2016) A reference vector guided evolutionary algorithm for many-objective optimization. IEEE Trans Evol Comput 20(5):773–791
57. Svozil D, Kvasnicka V, Pospichal J (1997) Introduction to multi-layer feed-forward neural networks. Chemom Intell Lab Syst 39(1):43–62
58. Hooke R, Jeeves TA (1961) “Direct search”solution of numerical and statistical problems. J ACM (JACM) 8(2):212–229
59. Tian Y, Cheng R, Zhang X, Jin Y (2017) Platemo: a matlab platform for evolutionary multi-objective optimization [educational forum]. IEEE Comput Intell Mag 12(4):73–87

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.