



A two-stage infill strategy and surrogate-ensemble assisted expensive many-objective optimization

Yi Zhao¹ · Jian Zhao¹ · Jianchao Zeng² · Ying Tan³

Received: 5 October 2021 / Accepted: 8 April 2022 / Published online: 3 May 2022
© The Author(s) 2022

Abstract

Many optimization problems are expensive in practical applications. The surrogate-assisted optimization methods have attracted extensive attention as they can get satisfyingly optimal solutions in a limited computing resource. In this paper, we propose a two-stage infill strategy and surrogate-ensemble assisted optimization algorithm for solving expensive many-objective optimization problems. In this method, the population is optimized by a surrogate ensemble. Then a two-stage infill strategy is proposed to select individuals for real evaluations. The infill strategy considers individuals with better convergence or greater uncertainty. To calculate the uncertainty, we consider two aspects. One is the approximate variance of the current surrogate ensemble and the other one is the approximate variance of the historical surrogate ensemble. Finally, the population is revised by the recently updated surrogate ensemble. In experiments, we testify our method on two sets of many-objective benchmark problems. The results demonstrate the superiority of our proposed algorithm compared with the state-of-the-art algorithms for solving computationally expensive many-objective optimization problems.

Keywords Surrogate ensemble · Many-objective optimization · Infill strategy · Expensive optimization problems

Introduction

In industrial optimization problems, such as electric vehicle control problems [4], industrial scheduling [12], and robotics [24], there are multiple objectives to be optimized simultaneously. These multi-objective optimization problems (MOPs) can be mathematically formulated as follows:

$$\text{minimize } F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})),$$

$$\text{subject to } \mathbf{x} \in R^D \quad (1)$$

where R^D represents the D -dimensional decision space, $f_i(\mathbf{x})$ represents the i th objective of individual \mathbf{x} and M is the number of objectives. Objectives are often conflicting with each other, and one objective getting better may cause another to deteriorate. Pareto optimal solutions obtained by multi-objective optimization algorithms can trade off the different objectives, and they are called Pareto set (PS) in the decision space and Pareto front (PF) in the objective space, respectively. Various methods like a fast and elitist multi-objective genetic algorithm (NSGA-II) [8], a multiobjective evolutionary algorithm based on decomposition (MOEA/D) [38], and indicator-based selection in multiobjective search (IBEA) [42] have been proposed for solving MOPs with two or three objectives. These multi-objective evolutionary algorithms (MOEAs) are confronted with the lack of selection pressure when the objectives increase. To solve many-objective optimization problems (MaOPs) with more than three objectives, a lot of methods have been proposed, for example, NSGA-II/SDR [31] based on the strengthened dominance relation (SDR), AR-MOEA [30] based on an IGD non contributing solution detection (IGD-NS) indicator, reference vector guided evolutionary algorithm (RVEA)

✉ Yi Zhao
yizhao1029@126.com

Jian Zhao
zhaojian7616@163.com

Jianchao Zeng
zjc@nuc.edu.cn

Ying Tan
tanying2015@163.com

¹ College of Mechanical Engineering, Taiyuan University of Science and Technology, Taiyuan 030024, China

² Department of Computer Science and Control Engineering, North University of China, Taiyuan 030051, China

³ Department of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China

[3] which decomposes a MOP into several single-objective optimization problems.

Usually, MOEAs require a lot of real evaluations of the objective function before finding a set of Pareto optimal solutions, which makes the MOEAs restricted in the expensive engineering problems, such as computing fluid kinetics (CFD) simulation [10] and engineering design optimization [11]. In these expensive problems, one simulation could take from minutes to hours [14]. Surrogate-assisted evolutionary algorithms (SAEAs), have been proposed to address the optimization of expensive problems. Commonly used surrogate models include polynomial regression model (PR) [12], radial basis function (RBF) [13], and Gaussian process (GP) [14]. For more descriptions of surrogate model, readers can refer to the review articles [13,14].

Surrogate-assisted evolutionary algorithms (SAEAs) can be divided into on-line and off-line optimization methods [33]. In the on-line surrogate-assisted optimization methods, a small number of expensive real fitness can be conducted during the optimization, and the newly generated samples can also be used to update the surrogates. Conversely, in the off-line surrogate-assisted optimization algorithms, there are not new samples available [35]. In the on-line surrogate-assisted optimization algorithms, the methods to select individuals for expensive evaluation, are also known as infill sampling criterion, infill strategy or surrogate management. Infill sampling criteria, such as expected improvement (EI) [21,25], lower confidence bound (LCB) [18], and probability of improvement (PoI) [9], are widely used in the Kriging or Gaussian process (GP) assisted optimization algorithms.

On-line SAEAs are more flexible than off-line SAEAs as they have additional samples for surrogate management during the optimization process, which may have more opportunities to improve the performance of the algorithm than off-line SAEAs [14]. Therefore, in this paper, our work mainly focuses on the on-line surrogate-assisted optimization methods. A lot of on-line surrogate-assisted single-objective optimization methods have been proposed. In [16], Li selected the individuals with the best approximated fitness and maximum uncertainty for expensive evaluations and used the distance and fitness value information to calculate the uncertainty. In [29], Tian proposed a multi-objective infill strategy that considers both approximated fitness and uncertainty for solving high-dimensional expensive problems. In [22], Pan used teaching-learning-based optimization (TLBO) or differential evolution (DE) alternately to search for the best candidate solution, and the generation-based and individual-based strategies are used for surrogate management. In [37], Yu used a GP model as a coarse surrogate to learn the global landscape and an RBF model as a fine surrogate to learn the local feature of the fitness landscape. In the coarse search, approximate fitness together with the uncertainty of GP is used for environmental selection. In [19], Liu

used the affinity propagation clustering technique to partition the population into several subpopulations and proposed an RBF-assisted learning strategy-based particle swarm optimizer (PSO) to update the particle in each subpopulation. For more methods of on-line single-objective SAEAs, please refer to Refs. [2,18,27,28,32,36].

Some on-line surrogate assisted-multi-objective optimization methods have been proposed. In ParEGO [15], a weight vector was selected at each iteration for optimization by the efficient global optimization (EGO). In MOEA/D-EGO [39], Gaussian process is built for each objective in the MOP and maximizing the expected improvement metrics are used to select test points for expensive evaluations. In KRVEA [5], the Kriging model is used to approximate each objective of MOPs, and approximated values or uncertainties provided by Kriging model are adaptively selected for expensive evaluations. In CSEA [23], Pan used a feedforward neural network (FNN) as a classifier to identify good solutions from the whole population. In [11], Guo used an efficient dropout neural network (EDN) to approximate the fitness of individuals and get their approximate uncertainties by randomly ignoring neurons in the neural network. In [34], Wang proposed an adaptive acquisition function in the Bayesian approach to solve expensive multi-objective optimization problems. In [26], Song used the two-archive evolutionary algorithm to optimize the population, and the differences between the individuals in the two archives are used for infill strategy. In [17], Lin used a global Kriging and several sub-models to construct the surrogate ensemble to approximate the objectives of the expensive multi-objective problems and proposed a reference vector-based infill strategy. In [10], Gu used the Kriging model to optimize the population for several generations, then the crowding degrees of the individuals in the radial space and the uncertainty information provided by the Kriging model are used for surrogate management. For ParEGO, MOEA/D-EGO, these two methods cannot obtain good performance on expensive MaOPs with more than three objectives. In KRVEA, the population is optimized by a set of uniformly distributed reference vectors, and the true evaluated individuals are selected from the population. When the shapes of reference vectors and PFs are not consistent with each other, the performance of the algorithm will be affected.

From the mentioned above, there have been some studies on multi-objective expensive optimization problems, but there are still some challenges in this field. First, different surrogate models are suitable for different types of expensive problems. Therefore, selecting an appropriate surrogate for the problems is important and difficult as there is no criterion for this work. Second, the multi-objective optimization framework also has an impact on the performance of the expensive MOPs. Third, as to surrogate management, infill sampling criteria directly affect the approximate accuracy of

the surrogate and the optimal solutions finally found by the methods.

On one hand, considering the first challenge of the expensive multi-objective optimization problems as described above, a natural idea is to combine multiple base learners to form a strong learner. And in our previous work [40], ensemble constructs a strong learner by combining multiple base learners, which has proved to be superior to a single learner in terms of accuracy and robustness. Furthermore, an ensemble surrogate can provide approximate variance among different surrogates, which is important for surrogate management. On the other hand, in SAEAs, the update process of surrogate model contains some historical information. At present, there is no method to use these historical information for surrogate management. Motivated by these, we propose a method to solve the expensive many-objective optimization problems, named a two-stage infill strategy and surrogate-ensemble assisted expensive many-objective evolutionary optimization algorithm (TSEMO). The main work of this paper consists of two aspects:

- (1) The RBF models based on two different kernel functions are used as the base learners to construct the surrogate ensemble. And the surrogate ensemble is used to optimize the population.
- (2) A two-stage infill strategy is proposed to select individuals for expensive function evaluations. In the first stage, the individuals with the minimum approximate fitness or the maximum uncertainty are selected for expensive evaluations. The uncertainty is the approximate variance of the surrogate ensemble. In the second stage, the individual with the maximum uncertainty is selected for expensive evaluations and the uncertainty is the approximate variance of the historical surrogate ensemble.

The rest of this paper is organized as follows. Section “Related work” introduces the relevant work. In “A two-stage infill strategy and surrogate-ensemble assisted expensive many-objective optimization”, the details of the proposed algorithm TSEMO are described. In “Numerical experiments”, the numerical experiments are carried out, and the results are compared with several methods on multi-objective benchmark problems. Finally, the conclusion and future work are drawn in “Conclusion”.

Related work

Different multi-objective optimization algorithms have a certain degree of impact on expensive multi-objective optimization problems. To solve the expensive many-objective optimization problems, we used NSGA-II/SDR [31] as the

underlying optimization framework of our method. And the NSGA-II/SDR algorithm is described below.

NSGA-II/SDR

Before giving NSGA-II/SDR, we first introduce the original dominant relation in NSGA-II, which is defined below.

Definition 1: an individual \mathbf{x} is said to dominate another individual \mathbf{y} (denoted as $\mathbf{x} < \mathbf{y}$) if and only if $\forall i, i = 1, 2, \dots, M, f_i(\mathbf{x}) \leq f_i(\mathbf{y})$, and there is at least one $j, j \in 1, 2, \dots, M$ satisfying $f_j(\mathbf{x}) < f_j(\mathbf{y})$.

Due to the lack of selection pressure, the traditional NSGA-II algorithm based on the dominant relation deteriorates in solving MaOPs with more than three objectives. To solve this problem, Tian [31] proposes a strengthened dominance relation (SDR), which replaces the original dominance relation in NSGA-II, and can greatly improve the selection pressure of many-objective optimization problems, thus promoting the performance of the algorithm. SDR can balance the convergence and diversity of MaOPs well by adopting the specific niche technology. The definition of strengthened dominance relation is given below.

Definition 2: an individual \mathbf{x} strengthened dominate another individual \mathbf{y} (denoted as $\mathbf{x} <_{SDR} \mathbf{y}$) if and only if

$$\begin{cases} \text{Con}(\mathbf{x}) < \text{Con}(\mathbf{y}) & \theta_{\mathbf{xy}} < \bar{\theta} \\ \text{Con}(\mathbf{x}) \cdot \frac{\theta_{\mathbf{xy}}}{\bar{\theta}} \leq \text{Con}(\mathbf{y}) & \theta_{\mathbf{xy}} \geq \bar{\theta} \end{cases} \quad (2)$$

where $\text{Con}(\mathbf{x}) = \sum_{i=1}^M f_i(\mathbf{x})$, $\theta_{\mathbf{xy}}$ denotes the acute angle between the two individual \mathbf{x}, \mathbf{y} in the objective space, $\bar{\theta}$ is the parameter which is set to the $\lfloor (|P|/2) \rfloor$ th minimum element of

$$\left\{ \min_{\mathbf{q} \in P \setminus \mathbf{p}} \theta_{\mathbf{pq}} \mid \mathbf{p} \in P \right\}. \quad (3)$$

SDR is irreflexive, antisymmetric, and nontransitive, respectively.

NSGA-II/SDR is based on the algorithm framework of NSGA-II. The multi-objective algorithms obtain a set of non-dominant solutions that are consistently distributed and proximate to the Pareto front by performing the environmental selection. During the environmental selection, NSGA-II/SDR first uses the strengthened dominance relation to sort the individuals into several non-dominant layers ($L_1, L_2, \dots, L_i, \dots$). We suppose that the parent population size is N , and the first $|i - 1|$ layers are put into the next generation, where $|L_1 \cup L_2 \cup \dots \cup L_{i-1}| < N$ and $|L_1 \cup L_2 \cup \dots \cup L_i| > N$. Crowding distance [8] which is the same as the original NSGA-II, is used to select the i th layer into the next generation one by one until the population size reaches N . More details of NSGA-II/SDR can be referred to [31].

A two-stage infill strategy and surrogate-ensemble assisted expensive many-objective optimization

Radial basis function(RBF) has been widely used in SAEAs. RBF fits well for nonlinear and higher-order problems and it has a low training complexity [41]. We use two different kernel functions, cubic and Gaussian function in RBF as the basic learners to construct the surrogate ensemble. The flowchart of TSEMO is given in Fig. 1 and the pseudocode of TSEMO is in Algorithm 1. In the main loop, TSEMO first uses Latin hypercube (LHS) [20] to generate $11D - 1$ individuals. These individuals are evaluated by the expensive functions and kept in archive A_1 . The non-dominated individuals in A_1 are kept in A_2 . We conduct the environmental selection of NSGA-II/SDR on the individuals in A_1 , and N individuals are selected as the initial population. Then the individuals in A_1 are used to train the surrogate ensemble for each objective of the MaOPs. After that, the population is optimized for w_{\max} generations. Next, a two-stage infill strategy is used to select individuals for expensive evaluations. These new samples are added to archive A_1 and used to update archive A_2 . The surrogate ensemble is re-trained by the samples in A_1 . At last, the population is revised with the surrogate ensemble. This process is repeated until the maximum number of expensive evaluations FE_{\max} is reached. In the subsequent subsections, we will give a detailed description of the methods on the surrogate ensemble-based optimization, choosing individuals for exact function evaluations (a two-stage infill strategy), and revising the population, respectively.

Surrogate-ensemble based optimization

In the surrogate-ensemble based optimization, all the individuals are evaluated by the RBF ensemble instead of the exact objective functions. Algorithm 2 gives the pseudocode of the surrogate-ensemble based optimization. In the surrogate-ensemble based optimization, we use simulated binary crossover [6] and polynomial mutation [7] to generate offspring population Q , and population Q is evaluated by RBF ensemble. The i th objective of individual \mathbf{x} is calculated below:

$$\tilde{f}_i(\mathbf{x}) = \frac{\tilde{f}_i^{\text{cubic}}(\mathbf{x}) + \tilde{f}_i^{\text{Gaussian}}(\mathbf{x})}{2} \quad (4)$$

where $\tilde{f}_i^{\text{cubic}}(\mathbf{x})$ and $\tilde{f}_i^{\text{Gaussian}}(\mathbf{x})$ denote i th objective of the individual \mathbf{x} evaluated by RBF models with cubic kernel function and Gaussian kernel function respectively. Then the objective functions of $P \cup Q$ are normalized by

$$F(\mathbf{x})' = \frac{F(\mathbf{x}) - F_{\min}}{F_{\max} - F_{\min}} \quad (5)$$

Algorithm 1 The pseudocode of TSEMO

Input: Maximum number of expensive function evaluations (FE_{\max}); population size N ;
Output: Solutions in A_2

- 1: Use LHS to generate the $11D - 1$ initial individuals and initialize the number of function evaluations $FE = 0$, set $A_1 = \emptyset$, $A_2 = \emptyset$, $A_3 = \emptyset$
- 2: Evaluate the initial individuals with the expensive objective functions, add them to A_1 , add all nondominated individuals of the samples into the archive A_2 ; and update $FE = 11D - 1$;
- 3: Select N individuals in A_1 as the initial population.
- 4: Use the samples in A_1 to train surrogate ensemble
- 5: **while** $FE < FE_{\max}$ **do**
- 6: Add the current population P into the archive A_3
- 7: Implement the surrogate-based optimization on the population P and get the population P' (**Refer to Algorithm 2**)
- 8: Add the population P' into the archive A_3
- 9: Select u individuals from the population P' by the two-stage infill strategy and re-evaluate them with the expensive functions (**Refer to Algorithm 3**)
- 10: Add u individuals obtained from step 9 to A_1 and update A_2
- 11: Update the surrogate ensemble with archive A_1
- 12: Implement the population revising strategy on the archive A_3 (**Refer to Algorithm 4**)
- 13: Reset the archive $A_3 = \emptyset$
- 14: **end while**

where F_{\max} and F_{\min} are the vectors of the maximum and minimum objective functions of population $P \cup Q$ respectively. Then the environmental selection of NSGA-II/SDR is used on the combined population $L = P \cup Q$ to select the next generation. If the maximum number of generation w_{\max} is reached, the surrogate-ensemble based optimization is ended and the population P' is as the output.

Algorithm 2 surrogate-ensemble based optimization

Input: population P
Output: population P'

- 1: **while** $w < w_{\max}$ **do**
- 2: Generate offspring Q using simulated binary crossover and polynomial mutation
- 3: Combine parent and offspring populations $L = P \cup Q$, and evaluate the objective values of L by RBF ensemble
- 4: Normalize the population L
- 5: Use environmental selection on L
- 6: Update $w = w + 1$
- 7: **end while**

A two-stage infill strategy

After the surrogate-ensemble assisted optimization, u individuals are selected for exact evaluations. In this paper, we propose a two-stage infill strategy for selecting individuals, and the pseudocode of the infill strategy is given in Algorithm 3.

In the first stage, the convergence of population or the uncertainty of the surrogate ensemble is adaptively con-

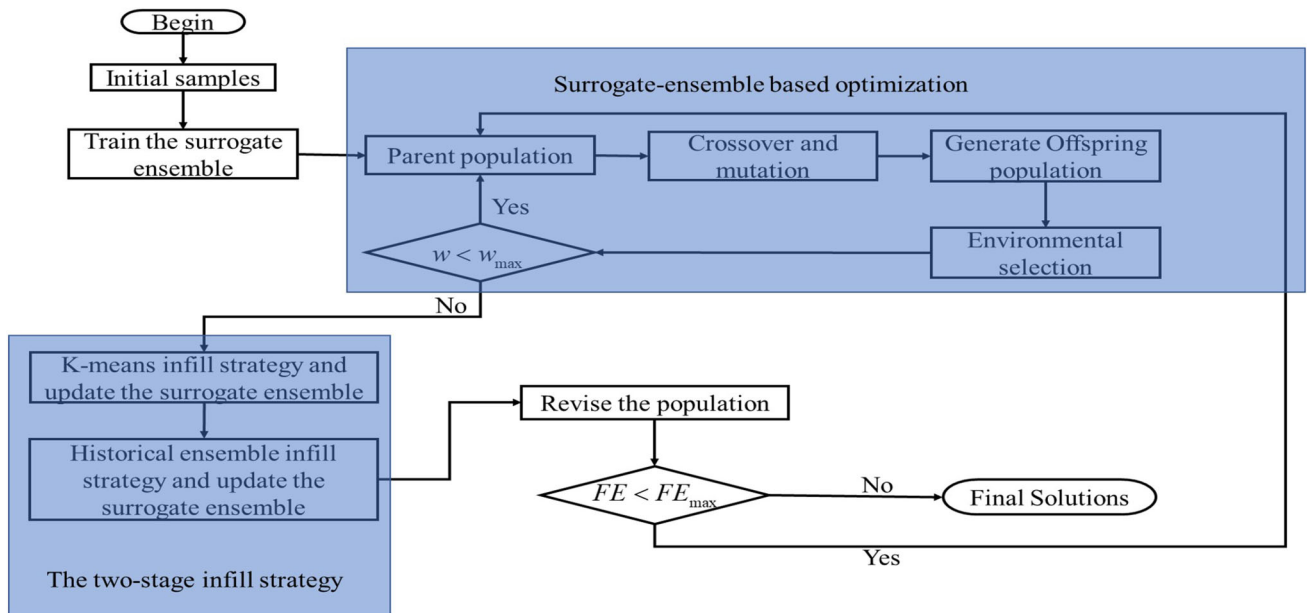


Fig. 1 The flow chart of TSEMO

sidered according to the distribution of the population. By considering the diversity of the population, we use k-means to divide the population P' into K clusters in objective space. An individual from each cluster with the minimum Euclidean distance (ED) to the origin or the maximum approximate variance ε_1 is selected for expensive function evaluations and ε_1 is calculated as follows:

$$\varepsilon_1(\mathbf{x}) = \frac{1}{M} \sum_{i=1}^M \left(\sqrt{(\tilde{f}_i^{\text{cubic}}(\mathbf{x}) - \tilde{f}_i^{\text{Gaussian}}(\mathbf{x}))^2} \right) \quad (6)$$

To select individuals from each cluster, the population P' is sorted into several non-dominated fronts by the original dominant relation in NSGA-II. For each cluster, if there is only one non-dominated front, which means the individuals are non-dominated with each other, the individual with the maximum uncertainty will be selected. Otherwise, the individual with the minimum Euclidean distance (ED) to the origin will be selected. Then the distance d_1 , which is the minimum distance from each of the K individuals to the samples in A_1 in the decision space is calculated. If the distance d_1 is greater than the threshold δ , the individual selected from the first stage will be evaluated by the exact objective function. After that, the individuals calculated by the expensive function are added to A_1 . Else if the distance d_1 is less than or equal to δ , the individual will not be reevaluated. Here we set the value of δ to 10^{-6} . A simple example is given in Fig. 2 to show the criterion to select individuals in each cluster. In Fig. 2, there are three clusters, c_1, c_2, c_3 , and one individual from each cluster will be selected. In cluster c_1 , because there are two non-dominated fronts, the individual a_1 with the small-

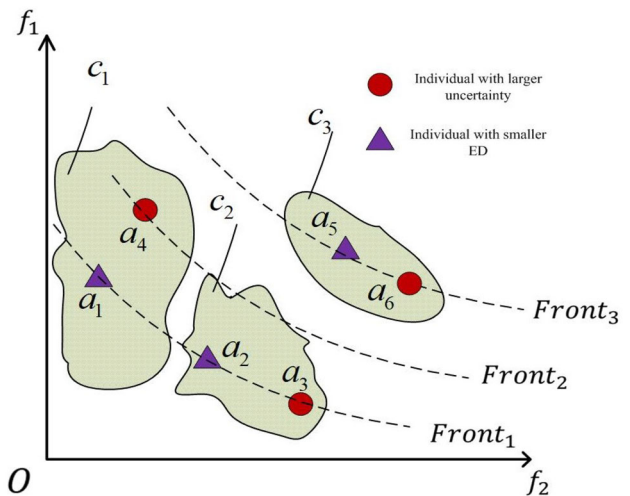


Fig. 2 An example of the infill strategy in the first stage

est ED value will be selected. In cluster c_2 , there is only one non-dominated front, thus the individual a_3 with the largest ε_1 will be selected. In cluster c_3 , a_6 will be selected as in cluster c_2 .

In the second stage, the $u - K$ individuals with the maximum uncertainty of ε_2 are selected. In this stage, the surrogate ensemble is updated by A_1 and the population P' is re-evaluated by the surrogate ensemble. The uncertainty ε_2 is calculated below:

$$\varepsilon_2(\mathbf{x}) = \frac{1}{M} \sum_{i=1}^M \left(\sqrt{(\tilde{f}_i^1(\mathbf{x}) - \tilde{f}_i^2(\mathbf{x}))^2} \right) \quad (7)$$

where $\tilde{f}_i^1(\mathbf{x})$ denotes the i th objective value of the individual \mathbf{x} approximated by the surrogate ensemble in the first stage and $\tilde{f}_i^2(\mathbf{x})$ denotes the i th objective value of the individual \mathbf{x} approximated by the surrogate ensemble in the second stage. The distance d_2 of $u - K$ individuals is calculated in the same way as d_1 . If d_2 is greater than the threshold δ , the individuals will be evaluated by the exact objective function and they will be added to A_1 . The approximated variance from the first and the second stage is used to select individuals for exact evaluation and the aim is to exploit the historical information of the surrogate ensemble.

Algorithm 3 A two-stage infill strategy

Input: population P' , archive A_1
Output: Set R_u

- 1: $(Front_1, Front_2, \dots, Front_L) = \text{Non-dominated-sort}(P')$
- 2: $(c_1, c_2, \dots, c_K) \leftarrow \text{Divide the population } P' \text{ into } K \text{ clusters using k-means method}$
- 3: **for** $i = 1$ to K **do**
- 4: Calculate the number of non-dominance layers τ in the c_i cluster
- 5: **if** $\tau = 1$ **then**
- 6: Select the individual with the maximum ε_1 value in each cluster
- 7: **else**
- 8: Select the individual with the minimum Euclidean distance to origin in each cluster
- 9: **end if**
- 10: Calculate the minimum distance d_1 from the individual to the samples in archive A_1
- 11: **if** $d_1 > \delta$ **then**
- 12: Evaluate the individual using the exact objective functions and add it to A_1 .
- 13: $FE = FE + 1$
- 14: **end if**
- 15: **end for**
- 16: Update the surrogate ensemble with archive A_1
- 17: Select $u - K$ individuals in P' with the maximum ε_2 value
- 18: **for** $j = 1$ to $u - K$ **do**
- 19: Calculate the minimum distance d_2 from the individual to the samples in archive A_1
- 20: **if** $d_2 > \delta$ **then**
- 21: Evaluate the individual using the exact objective functions and add it to A_1 .
- 22: $FE = FE + 1$
- 23: **end if**
- 24: **end for**

Revising the population

In surrogate-assisted optimization, the approximated errors of objectives may lead to the wrong direction of search. And the approximated objectives of the surrogate ensemble may become increasingly accurate as the training samples in A_1 increase. Considering the above two points, we propose a strategy to revise the population. First, the population P before the surrogate-based optimization and the population P' are combined into A_3 . After the two-stage infill strategy,

the surrogate ensemble will be updated by the samples in A_1 . Then the objectives of the individuals in A_3 are re-evaluated by the updated surrogate ensemble. Finally, environmental selection of NSGA-II/SDR is performed on A_3 and N individuals with better convergence and diversity are retained. Figure 3 gives an example to explain the effect of revising the population. Figure 3 shows a two-dimensional decision space. In Fig. 3, region 1 represents the region occupied by individuals with worse convergence and smaller approximate error, region 2 represents the region occupied by individuals with better convergence and larger approximate error, region 3 represents the region occupied by individuals with better convergence and smaller approximate error, and region 4 represents the region occupied by individuals with worse convergence and larger approximate error. The five-pointed star represents the individuals whose exact function values are convergent. In the process of surrogate-ensemble based optimization, the optimization of the population is based on approximated value of the surrogate ensemble. Besides, due to the limited training samples at the initial stage of optimization, approximated error of the surrogate ensemble is large, the individuals in region 4 with worse convergence will be discarded. However, region 4 may contain some individuals with good convergence. By revising the population, the initial population is re-evaluated with the updated surrogate ensemble after the two-stage infill strategy. At this time, the approximate accuracy of the surrogate ensemble is improved due to the increased samples. In this way, the individuals represented by the five-pointed star in region 4 can be retained. And the effectiveness of this step will be validated in the experimental section.

Algorithm 4 Revising the population

Input: The archive A_3
Output: Population P

- 1: Evaluate the objective values of the individuals in the archive A_3 by RBF ensemble
- 2: $P = \text{Environmental_Selection}(A_3)$

Numerical experiments

In this section, we first describe the performance metrics and parameter settings. Then we test the performance of TSEMO by comparing it with several state-of-the-art MaOPs algorithms on DTLZ and MaF test problems. After that, we conduct experiments on different values of K to determine the thresholds used in the two-stage infill strategy in TSEMO. Then the effectiveness of the two-stage infill strategy is studied. At last, the efficiency of revising the population is examined. Wilcoxon rank-sum test is used to compare the

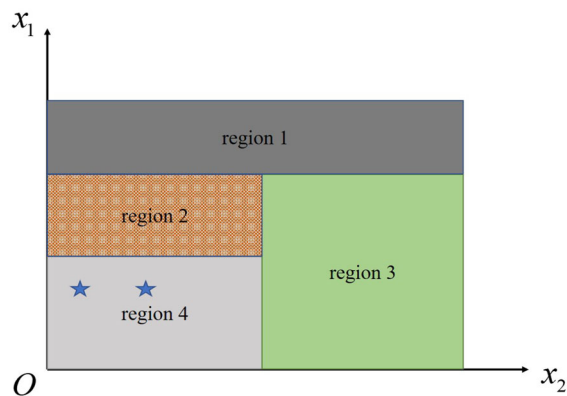


Fig. 3 An example to explain the effect of revising the population

results of different algorithms and each algorithm is run 20 times independently. The significance level is set to 0.05. And the symbol ‘+’ indicates that the compared algorithm performs statistically better than TSEMO. The ‘-’ represents that the compared algorithm performs statistically worse than TSEMO. And the ‘≈’ means that there is no significant difference between the performance of TSEMO and the compared algorithm.

Performance metrics and experimental settings

In this paper, inverted generational distance (IGD) [1] is used as the performance indicator to compare the performance of different algorithms. The IGD indicator can provide information about convergence and diversity of the non-dominated solutions, and it is defined as follows:

$$IGD(P^*, \Omega) = \frac{\sum_{x \in P^*} \text{dis}(x, P^*)}{|P^*|} \tag{8}$$

where P^* denotes the evenly distributed point set on the true PF, Ω is the non-dominated solutions obtained by the algorithm and $\text{dis}(x, P^*)$ represents the minimum Euclidean distance between x and the point in P^* . The number of the reference points in the true PF is set as recommended in KRVEA [5].

We also use HV as the performance indicator to compare the performance of our method with other algorithms. The HV indicator measures the volume of the hypercube dominated by the non-dominated solutions obtained by the algorithm, and it is defined below:

$$HV(\Omega | \mathbf{z}^r) = \text{Vol} \left(\bigcup_{\mathbf{x} \in \Omega} [f_1(\mathbf{x}), z_1^r] \times [f_2(\mathbf{x}), z_2^r] \times \dots \times [f_M(\mathbf{x}), z_M^r] \right) \tag{9}$$

where $\text{Vol}(\cdot)$ indicates the Lebesgue measure, and $\mathbf{z}^r = (z_1^r, z_2^r, \dots, z_M^r)$ is a reference point in the objective space that is dominated by Ω . \mathbf{z}^r is set as recommended in ARMOEA [30].

The experimental settings are given below:

- (1) Genetic operators: in this paper, the simulated binary crossover [6] and polynomial mutation [7] are used to generate offspring population in all the algorithms. The distribution index of crossover is set to 20, and the distribution index of mutation is set to 20. The crossover probability p_c is set to 1.0, and the mutation probability p_m is set to $1/D$.
- (2) Population size: the population of TSEMO is set to 50 and the compared algorithms are set as their corresponding references.
- (3) The termination condition: the maximum number of function evaluations is used as the termination condition and it is set to 300.
- (4) The number of evolutionary generations assisted by surrogate-ensemble w_{\max} is set to 20 as the empirical value in KRVEA [5].

Experimental results

In this section, we compare TSEMO with three surrogate-assisted algorithms namely, KTA2, CSEA, KRVEA, and two MOEAs called NSGA-II/SDR, RVEA on DTLZ1-DTLZ7, and the comparison results are given in Tables 1 and 2. The objective dimensions of DTLZ problems are 3, 4, 6, 8, 10. We also test the performance of TSEMO and other compared algorithms on MaF1–MaF7. The objective dimensions of MaF problems are 3, 5, 10. The decision variable dimensions of the two test suites are 10.

The statistical results of IGD in Table 1 show that TSEMO obtained the best results on all the test problems of DTLZ2, DTLZ5, and DTLZ6, and TSEMO obtained the best results on most of the test problems of DTLZ4 and DTLZ7. For two multi-objective algorithms, NSGA-II/SDR and RVEA, TSEMO obtained 25 better results than NSGA-II/SDR and RVEA among 35 test problems, which demonstrate that surrogate models are effective on most test problems. For a few test problems like DTLZ1 and DTLZ3, TSEMO needs more expensive function evaluations to find the true PF. TSEMO got 20 better results and 12 worse results than KTA2. TSEMO got 18 better results and 6 worse results than KRVEA. Compared to CSEA, TSEMO obtained 23 better results and 11 worse results. Then the statistical results of HV in Table 2 also show that TSEMO performed the best among all the algorithms. Therefore, for DTLZ test suite, TSEMO is more efficient than the three representative surrogate-assisted many-objective optimization algorithms, KTA2, KRVEA, and CSEA. Next, the parallel coordinates plot of the final

Table 1 Statistical results for IGD values obtained by KTA2, KRVEA, CSEA, NSGA-II/SDR, RVEA and TSEMO

Problem	M	KTA2	KRVEA	CSEA	NSGAII/SDR	RVEA	TSEMO
DTLZ1	3	4.8197e+1 (1.41e+1) +	8.6476e+1 (2.13e+1) ≈	5.9534e+1 (1.25e+1) +	8.7453e+1 (1.88e+1) ≈	7.7741e+1 (1.70e+1) ≈	8.2409e+1 (1.61e+1)
	4	3.6212e+1 (9.26e+0) +	5.5454e+1 (1.35e+1) ≈	4.0939e+1 (1.13e+1) +	6.5287e+1 (1.91e+1) ≈	4.4092e+1 (1.36e+1) +	6.4502e+1 (1.28e+1)
	6	1.4708e+1 (5.22e+0) +	2.6450e+1 (8.06e+0) +	1.5112e+1 (4.53e+0) +	2.3110e+1 (7.58e+0) +	1.6412e+1 (5.87e+0) +	3.0229e+1 (5.41e+0)
	8	5.8784e+0 (2.25e+0) +	6.4006e+0 (2.91e+0) +	4.2113e+0 (2.20e+0) +	5.7819e+0 (2.02e+0) +	5.5389e+0 (2.81e+0) +	9.1380e+0 (3.76e+0)
	10	3.2361e−1 (8.38e−2) +	2.9790e−1 (7.57e−2) +	2.8425e−1 (8.21e−2) +	3.7795e−1 (1.43e−1) ≈	4.6997e−1 (4.49e−1) ≈	4.0968e−1 (1.18e−1)
	DTLZ2	3	6.4847e−2 (4.08e−3) −	1.2425e−1 (2.06e−2) −	2.4099e−1 (3.53e−2) −	2.8750e−1 (2.42e−2) −	2.8741e−1 (3.19e−2) −
4		1.4633e−1 (5.96e−3) −	2.1435e−1 (1.76e−2) −	3.2764e−1 (3.60e−2) −	3.3927e−1 (3.33e−2) −	3.4220e−1 (3.57e−2) −	1.2823e−1 (1.99e−3)
6		2.9517e−1 (1.16e−2) −	3.4479e−1 (1.31e−2) −	4.9143e−1 (3.25e−2) −	4.4469e−1 (3.55e−2) −	4.4625e−1 (2.88e−2) −	2.7141e−1 (3.40e−3)
8		4.0115e−1 (8.47e−3) −	4.4077e−1 (1.26e−2) −	6.2001e−1 (3.29e−2) −	6.1542e−1 (5.04e−2) −	5.6864e−1 (3.87e−2) −	3.8555e−1 (5.97e−3)
10		4.3593e−1 (7.90e−3) +	5.2457e−1 (2.40e−2) −	6.7637e−1 (2.34e−2) −	7.1281e−1 (3.13e−2) −	7.6130e−1 (1.03e−1) −	5.0242e−1 (3.23e−2)
DTLZ3		3	1.2253e+2 (4.87e+1) +	2.1202e+2 (5.29e+1) ≈	1.5569e+2 (3.37e+1) +	2.3380e+2 (6.28e+1) ≈	2.1505e+2 (6.18e+1) ≈
	4	1.0640e+2 (3.22e+1) +	1.7845e+2 (2.85e+1) ≈	1.1189e+2 (3.87e+1) +	1.7309e+2 (4.84e+1) ≈	1.5149e+2 (3.67e+1) ≈	1.7526e+2 (3.77e+1)
	6	5.0164e+1 (1.57e+1) +	8.3980e+1 (2.79e+1) ≈	5.9675e+1 (1.58e+1) +	8.4158e+1 (2.12e+1) +	6.2118e+1 (1.78e+1) +	1.0040e+2 (2.87e+1)
	8	1.6355e+1 (1.02e+1) +	2.3146e+1 (1.02e+1) +	1.4747e+1 (5.19e+0) +	2.1417e+1 (1.01e+1) +	2.0582e+1 (9.33e+0) +	3.2472e+1 (1.04e+1)
	10	1.1251e+0 (3.67e−1) +	1.0072e+0 (2.26e−1) +	9.5269e−1 (1.24e−1) +	1.2273e+0 (3.18e−1) +	2.3365e+0 (2.80e+0) ≈	1.5560e+0 (5.59e−1)
	DTLZ4	3	3.4258e−1 (2.57e−1) ≈	3.2302e−1 (9.81e−2) ≈	4.6070e−1 (1.75e−1) −	8.1075e−1 (1.93e−1) −	4.1604e−1 (1.36e−1) −
4		4.1919e−1 (1.41e−1) ≈	3.9873e−1 (7.72e−2) ≈	3.3243e−1 (5.81e−2) +	8.9105e−1 (1.50e−1) −	6.0232e−1 (1.92e−1) −	3.9368e−1 (5.14e−2)
6		5.5366e−1 (1.02e−1) −	5.0307e−1 (5.30e−2) ≈	4.7341e−1 (4.64e−2) ≈	1.0145e+0 (1.23e−1) −	6.9464e−1 (1.16e−1) −	4.9458e−1 (3.41e−2)
8		5.8830e−1 (8.09e−2) −	5.4137e−1 (2.63e−2) ≈	5.7000e−1 (3.27e−2) −	1.0273e+0 (1.72e−1) −	7.5990e−1 (1.12e−1) −	5.2261e−1 (2.83e−2)
10		5.9165e−1 (4.09e−2) −	6.4033e−1 (4.01e−2) −	6.4429e−1 (3.06e−2) −	8.0551e−1 (9.02e−2) −	9.4315e−1 (8.41e−2) −	5.6244e−1 (2.05e−2)
DTLZ5		3	1.0771e+2 (2.88e−3) +	7.1794e−2 (1.81e−2) −	1.2398e−1 (3.43e−2) −	2.1726e−1 (2.21e−2) −	2.3387e−1 (3.85e−2) −
	4	5.5606e−2 (1.32e−2) −	5.7711e−2 (1.83e−2) −	1.2978e−1 (2.21e−2) −	1.9692e−1 (2.86e−2) −	2.2688e−1 (6.44e−2) −	3.1403e−2 (6.38e−3)
	6	6.7885e−2 (1.91e−2) −	3.8141e−2 (1.10e−2) −	8.0606e−2 (2.01e−2) −	1.4564e−1 (2.26e−2) −	2.5829e−1 (6.01e−2) −	2.8630e−2 (4.36e−3)
	8	6.3113e−2 (1.37e−2) −	2.4486e−2 (5.31e−3) −	4.6715e−2 (8.53e−3) −	1.1937e−1 (2.95e−2) −	1.7701e−1 (6.49e−2) −	1.9254e−2 (2.43e−3)
	10	1.9667e−2 (3.37e−3) −	1.2780e−2 (1.99e−3) −	1.4326e−2 (1.36e−3) −	7.9427e−2 (1.73e−2) −	2.3577e−1 (1.55e−1) −	1.0464e−2 (1.71e−3)
	DTLZ6	3	1.7679e+0 (5.03e−1) ≈	2.7377e+0 (3.13e−1) −	5.8812e+0 (4.59e−1) −	5.9324e+0 (3.22e−1) −	5.8139e+0 (3.44e−1) −
4		1.8498e+0 (5.75e−1) −	1.8977e+0 (3.90e−1) −	5.0638e+0 (4.86e−1) −	5.0459e+0 (3.89e−1) −	5.1390e+0 (3.77e−1) −	1.4274e+0 (4.77e−1)
6		1.4152e+0 (5.07e−1) −	1.4293e+0 (3.33e−1) −	3.0018e+0 (4.85e−1) −	3.5580e+0 (4.02e−1) −	3.5133e+0 (4.36e−1) −	8.7852e−1 (3.46e−1)
8		9.9739e−1 (4.33e−1) −	5.9535e−1 (2.39e−1) −	1.2243e+0 (4.47e−1) −	1.8905e+0 (2.39e−1) −	1.8230e+0 (5.04e−1) −	3.6345e−1 (9.95e−2)
10		9.6057e−2 (3.86e−2) −	9.1568e−2 (3.51e−2) −	1.9395e−1 (2.37e−1) −	5.1643e−1 (2.44e−1) −	6.0775e−1 (2.53e−1) −	4.6694e−2 (1.67e−2)
DTLZ7		3	4.2266e−1 (2.78e−1) −	1.0584e−1 (8.47e−3) +	1.4927e+0 (4.91e−1) −	4.6140e+0 (1.10e+0) −	3.3389e+0 (8.53e−1) −
	4	6.1968e−1 (3.05e−1) −	2.4522e−1 (2.19e−2) ≈	2.8625e+0 (9.00e−1) −	5.5256e+0 (1.13e+0) −	3.5830e+0 (1.15e+0) −	2.4455e−1 (2.21e−2)
	6	6.7099e−1 (2.09e−1) −	5.4475e−1 (7.09e−2) −	8.5548e+0 (2.43e+0) −	5.6257e+0 (2.17e+0) −	4.3386e+0 (2.20e+0) −	5.0508e−1 (2.38e−2)
	8	1.0837e+0 (3.42e−1) −	8.5052e−1 (3.50e−2) −	8.2860e+0 (2.97e+0) −	4.2442e+0 (1.91e+0) −	4.0890e+0 (2.53e+0) −	8.1673e−1 (4.02e−2)
	10	1.3457e+0 (3.41e−1) −	1.0179e+0 (3.10e−2) +	2.0119e+0 (2.98e−1) −	1.7050e+0 (6.41e−1) −	2.0806e+0 (5.05e−1) −	1.1391e+0 (2.24e−2)
	+ / − / ≈	12/20/3	6/18/11	11/23/1	5/25/5	5/25/5	5/25/5

The best results are highlighted

Table 2 Statistical results for HV values obtained by KTA2, KRVEA, CSEA, NSGA-II/SDR, RVEA and TSEMO

Problem	M	KTA2	KRVEA	CSEA	NSGAII/SDR	RVEA	TSEMO
DTLZ1	3	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)
	4	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)
DTLZ2	6	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)
	8	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)
DTLZ3	10	4.5565e-1 (2.05e-1) +	5.3965e-1 (2.12e-1) +	6.2029e-1 (2.37e-1) +	3.3261e-1 (1.95e-1) ≈	3.4151e-1 (2.40e-1) ≈	2.4223e-1 (1.40e-1)
	3	5.3472e-1 (7.23e-3) -	4.5092e-1 (3.24e-2) -	2.5851e-1 (6.31e-2) -	1.4936e-1 (2.85e-2) -	1.5675e-1 (4.45e-2) -	5.5881e-1 (1.84e-3)
DTLZ4	4	6.2954e-1 (1.28e-2) -	5.8793e-1 (2.87e-2) -	3.1312e-1 (6.37e-2) -	2.3207e-1 (5.42e-2) -	2.2437e-1 (5.57e-2) -	6.9665e-1 (1.83e-3)
	6	6.6748e-1 (3.24e-2) -	6.9278e-1 (3.82e-2) -	4.3460e-1 (4.65e-2) -	3.7987e-1 (5.45e-2) -	3.3131e-1 (4.53e-2) -	8.3059e-1 (6.72e-3)
DTLZ5	8	6.6583e-1 (2.94e-2) -	8.2332e-1 (1.52e-2) -	5.4443e-1 (2.88e-2) -	4.4937e-1 (4.01e-2) -	4.4184e-1 (4.42e-2) -	8.9799e-1 (6.59e-3)
	10	7.8976e-1 (4.65e-2) -	7.5660e-1 (1.90e-2) -	6.3300e-1 (2.24e-2) -	5.7113e-1 (3.54e-2) -	3.1803e-1 (3.93e-2) -	9.3054e-1 (1.23e-2)
DTLZ6	3	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)
	4	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)
DTLZ7	6	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)
	8	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)
DTLZ8	10	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)
	3	1.7538e-1 (1.31e-1) +	1.7055e-1 (1.24e-1) +	2.9350e-1 (1.23e-1) +	7.1228e-2 (9.70e-2) +	1.5778e-1 (1.24e-1) +	5.3933e-3 (1.67e-2)
DTLZ9	4	3.1439e-1 (1.06e-1) +	1.6139e-1 (1.00e-1) -	2.0314e-1 (6.50e-2) -	4.3764e-2 (3.50e-2) -	1.2960e-1 (4.20e-2) -	2.5720e-1 (6.80e-2)
	6	3.7245e-1 (1.10e-1) +	2.0713e-1 (1.27e-1) -	4.1776e-1 (7.29e-2) +	7.8793e-2 (5.11e-2) -	1.9356e-1 (9.27e-2) -	3.1667e-1 (8.05e-2)
DTLZ10	8	4.4132e-1 (9.85e-2) -	4.0728e-1 (1.27e-1) -	6.0523e-1 (4.73e-2) +	1.3230e-1 (5.47e-2) -	3.1889e-1 (1.12e-1) -	4.8187e-1 (7.18e-2)
	10	6.1538e-1 (1.21e-1) -	6.6106e-1 (7.55e-2) -	6.9543e-1 (4.22e-2) ≈	2.1365e-1 (1.32e-1) -	4.4734e-1 (1.06e-1) -	7.1431e-1 (4.48e-2)
DTLZ11	3	8.3660e-1 (6.01e-2) ≈	7.7843e-1 (6.16e-2) -	7.9083e-1 (4.09e-2) -	5.6400e-1 (1.26e-1) -	3.8251e-1 (8.43e-2) -	8.6871e-1 (2.45e-2)
	4	1.9470e-1 (1.59e-3) +	1.3957e-1 (1.50e-2) -	8.4038e-2 (3.11e-2) -	2.8485e-2 (8.85e-3) -	2.4297e-2 (1.68e-2) -	1.8483e-1 (3.09e-3)
DTLZ12	6	1.1047e-1 (1.74e-2) -	1.2500e-1 (1.11e-2) -	5.9889e-2 (2.06e-2) -	1.8965e-2 (1.62e-2) -	3.4806e-2 (2.51e-2) -	1.3543e-1 (2.25e-3)
	8	6.0436e-2 (2.90e-2) -	1.0644e-1 (3.85e-3) ≈	8.9282e-2 (1.04e-2) -	1.6157e-2 (1.13e-2) -	3.6481e-2 (2.92e-2) -	1.0801e-1 (1.44e-3)
DTLZ13	10	5.7366e-2 (2.52e-2) -	9.8821e-2 (1.61e-3) -	9.8043e-2 (3.86e-3) -	4.3551e-2 (2.67e-2) -	7.1807e-2 (2.64e-2) -	1.0051e-1 (9.49e-4)
	3	8.0230e-2 (1.35e-2) -	9.7530e-2 (7.27e-4) -	9.9604e-2 (4.94e-4) +	6.2491e-2 (2.46e-2) -	9.1838e-2 (2.11e-3) -	9.8524e-2 (6.85e-4)
DTLZ14	4	2.4393e-5 (1.09e-4) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)
	6	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)
DTLZ15	8	3.3237e-7 (1.49e-6) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0) ≈	0.0000e+0 (0.00e+0)
	10	4.0956e-6 (1.83e-5) -	1.6013e-2 (3.36e-2) ≈	3.0666e-6 (1.37e-5) -	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	3.1046e-7 (1.39e-6)
DTLZ16	3	6.1662e-2 (4.04e-2) -	9.5072e-2 (1.10e-3) ≈	7.5335e-2 (3.89e-2) ≈	8.1519e-3 (2.09e-2) -	2.4280e-2 (4.03e-2) -	9.2779e-2 (8.52e-3)
	4	2.3741e-1 (2.78e-2) ≈	2.5893e-1 (3.28e-3) +	4.3238e-2 (3.29e-2) -	4.1895e-6 (1.87e-5) -	9.4449e-5 (4.22e-4) -	2.3256e-1 (9.32e-3)
DTLZ17	6	2.3788e-1 (1.86e-2) +	2.3652e-1 (1.00e-2) +	1.7707e-2 (2.27e-2) -	0.0000e+0 (0.00e+0) -	4.1339e-4 (1.20e-3) -	2.2165e-1 (7.96e-3)
	8	2.1718e-1 (1.05e-2) +	2.1042e-1 (6.40e-3) +	2.0946e-3 (9.37e-3) -	3.5213e-5 (1.06e-4) -	9.7638e-4 (2.31e-3) -	1.7979e-1 (1.73e-2)
DTLZ18	10	1.8852e-1 (1.51e-2) +	1.8332e-1 (4.44e-3) +	3.3476e-4 (1.47e-3) -	1.2457e-3 (1.63e-3) -	1.9651e-3 (3.09e-3) -	5.7619e-2 (3.98e-2)
	+/ - / ≈	9/13/13	7/14/14	6/16/13	1/2/1/13	1.0074e-2 (1.28e-2) +	3.5410e-3 (3.84e-3)

The best results are highlighted

Table 3 Statistical results for IGD values obtained by KTA2, KRVEA, CSEA, NSGA-III/SDR, RVEA and TSEMO

Problem	M	KRVEA	CSEA	NSGAIII/SDR	RVEA	TSEMO
MaF1	3	5.6797e-2 (4.55e-3) -	1.7663e-1 (2.58e-2) -	2.9460e-1 (3.65e-2) -	2.4075e-1 (4.25e-2) -	3.8212e-2 (1.48e-3)
	5	1.4934e-1 (2.43e-2) -	2.1933e-1 (1.83e-2) -	3.4144e-1 (4.02e-2) -	3.7000e-1 (4.98e-2) -	1.1410e-1 (1.76e-3)
	10	4.1587e-1 (2.02e-2) -	2.6136e-1 (1.30e-2) +	3.5774e-1 (2.51e-2) -	6.1429e-1 (8.43e-2) -	3.0831e-1 (2.93e-2)
MaF2	3	3.8116e-2 (1.33e-3) -	5.1787e-2 (2.55e-3) -	5.6111e-2 (2.66e-3) -	7.3221e-2 (4.11e-3) -	2.9108e-2 (8.58e-4)
	5	9.0676e-2 (1.87e-3) -	9.9174e-2 (2.05e-3) -	1.3037e-1 (4.56e-3) -	1.5663e-1 (6.79e-3) -	8.9262e-2 (8.56e-4)
	10	3.0575e-1 (2.56e-2) -	3.3337e-1 (1.75e-2) -	4.3777e-1 (5.99e-2) -	4.8867e-1 (1.05e-1) -	2.7365e-1 (1.47e-2)
MaF3	3	5.7236e+5 (2.75e+5) -	2.2830e+5 (2.27e+5) \approx	1.3020e+5 (6.83e+4) \approx	1.3028e+5 (9.09e+4) \approx	1.2695e+5 (5.64e+4)
	5	2.2184e+5 (1.19e+5) -	1.1625e+5 (1.01e+5) \approx	3.3553e+4 (2.05e+4) +	6.7983e+4 (5.09e+4) \approx	6.3356e+4 (3.68e+4)
	10	3.6928e+0 (4.82e+0) \approx	9.9204e-1 (1.59e+0) +	1.7695e+0 (1.42e+0) \approx	4.1156e+0 (7.84e+0) \approx	3.5757e+0 (5.30e+0)
MaF4	3	8.1020e+2 (2.73e+2) \approx	5.4813e+2 (1.29e+2) +	9.7838e+2 (2.10e+2) -	6.6785e+2 (1.38e+2) \approx	7.7610e+2 (1.78e+2)
	5	1.2016e+3 (3.77e+2) +	1.2612e+3 (3.83e+2) +	1.8689e+3 (5.52e+2) \approx	1.4177e+3 (5.14e+2) +	2.1746e+3 (5.39e+2)
	10	2.6880e+2 (8.44e+1) +	2.1824e+2 (9.47e+1) +	3.7203e+2 (2.72e+2) +	9.8421e+2 (1.42e+3) \approx	6.4431e+2 (2.46e+2)
MaF5	3	1.3278e+0 (3.67e-1) \approx	1.5802e+0 (5.22e-1) \approx	3.0502e+0 (1.21e+0) -	1.8699e+0 (5.46e-1) -	1.3352e+0 (1.63e-1)
	5	4.5402e+0 (5.28e-1) +	4.5112e+0 (1.00e+0) +	1.1297e+1 (5.09e+0) -	7.0364e+0 (1.56e+0) -	5.8641e+0 (6.77e-1)
	10	1.0870e+2 (2.27e+1) \approx	1.1241e+2 (2.08e+1) \approx	2.1780e+2 (7.52e+1) -	1.9649e+2 (2.43e+1) -	1.1346e+2 (1.93e+1)
MaF6	3	9.1551e-1 (4.61e-1) -	3.5913e+0 (2.07e+0) -	1.1417e+1 (4.53e+0) -	7.7737e+0 (3.18e+0) -	8.2398e-2 (2.05e-2)
	5	3.2016e-1 (1.07e-1) -	1.5907e+0 (6.19e-1) -	4.0576e+0 (2.35e+0) -	4.1643e+0 (2.56e+0) -	2.4117e-1 (5.67e-2)
	10	1.8401e-2 (8.91e-3) +	2.3343e-2 (7.59e-3) +	6.7844e-2 (3.47e-2) \approx	2.7905e-1 (1.89e-1) -	4.9090e-2 (1.90e-2)
MaF7	3	1.0756e-1 (1.11e-2) +	1.4357e+0 (7.20e-1) -	4.4204e+0 (9.04e-1) -	2.8851e+0 (9.53e-1) -	1.4857e-1 (1.58e-2)
	5	4.5211e-1 (1.05e-1) -	5.1999e+0 (1.41e+0) -	6.0043e+0 (1.62e+0) -	4.8400e+0 (1.22e+0) -	3.3673e-1 (1.28e-2)
	10	1.0302e+0 (3.80e-2) +	1.9913e+0 (2.59e-1) -	1.8785e+0 (5.34e-1) -	1.8791e+0 (5.07e-1) -	1.1490e+0 (2.87e-2)
+ / - / \approx	6/11/4	7/10/4	2/15/4	1/15/5		

The best results are highlighted

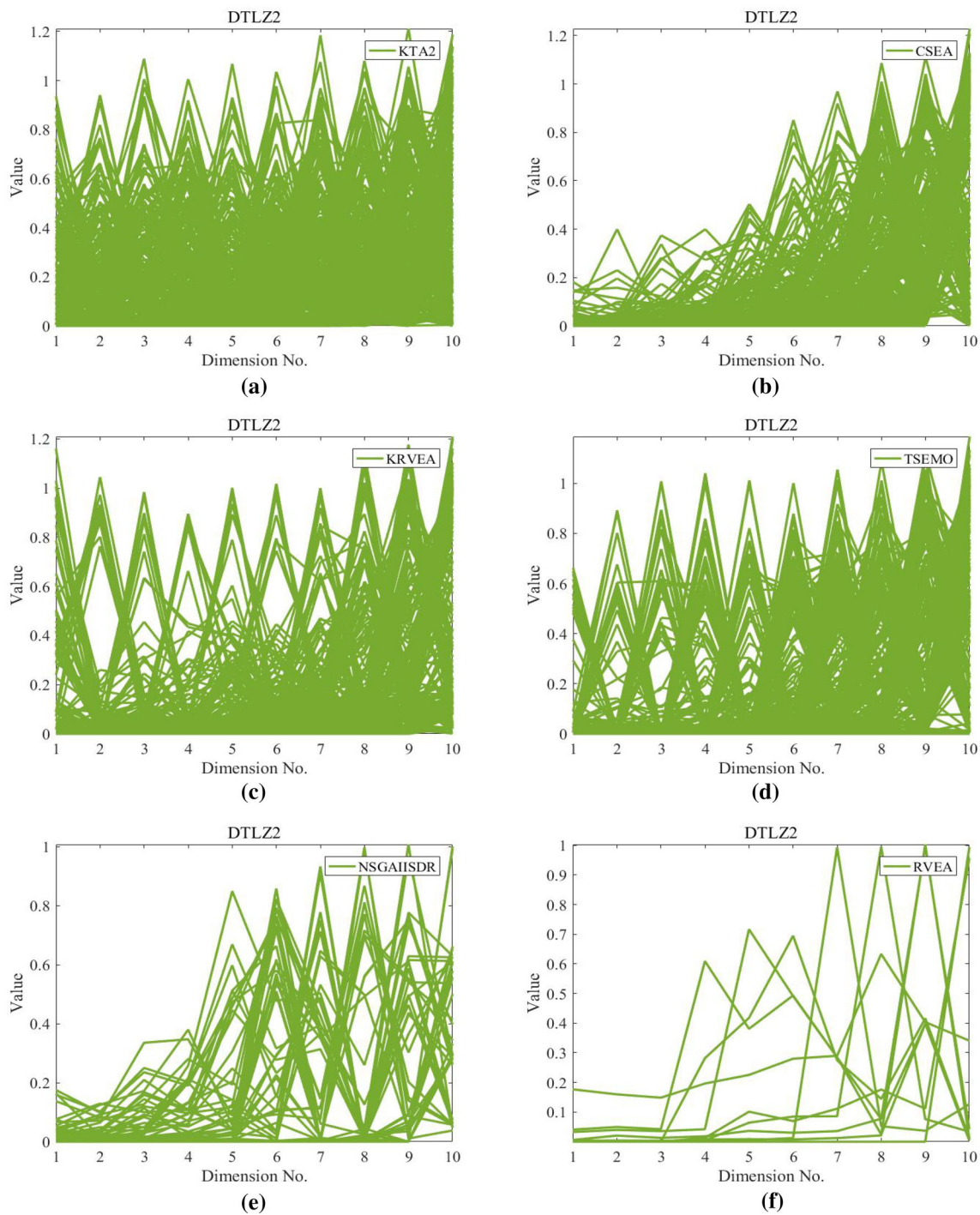


Fig. 4 Parallel coordinate plot of the nondominated solutions obtained by KTA2, CSEA, K-RVEA, TSEMO, NSGA-II/SDR, and RVEA in the run with the best IGD value on the 10-objective DTLZ2

non-dominated solutions obtained by TSEMO and the compared algorithms for DTLZ2 with 10 objectives is presented in Fig. 4. From it, we can see that the maximum value of objectives for all the non-dominated solutions obtained by TSEMO is smaller than that of the compared algorithms and the non-dominated solutions of TSEMO have a higher den-

sity than the two non-surrogate assisted MOEAs algorithms. These mean TSEMO can achieve a set of well convergent and evenly distributed solutions and has a better capability on the convergence and diversity. As to MOEAs, NSGA-II/SDR and RVEA have a lower density of solutions which means they have a worse distribution.

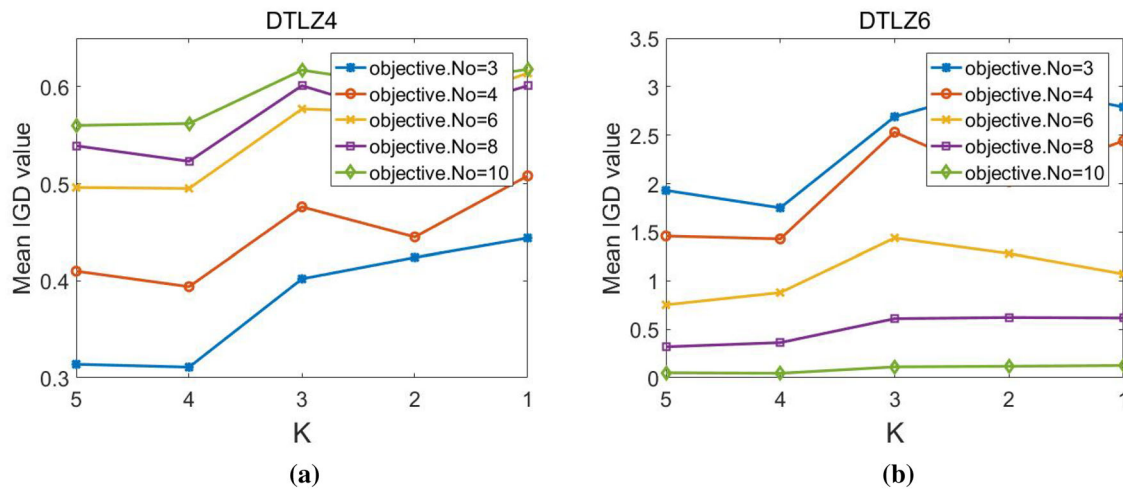


Fig. 5 The mean IGD values obtained by TSEMO with different threshold K

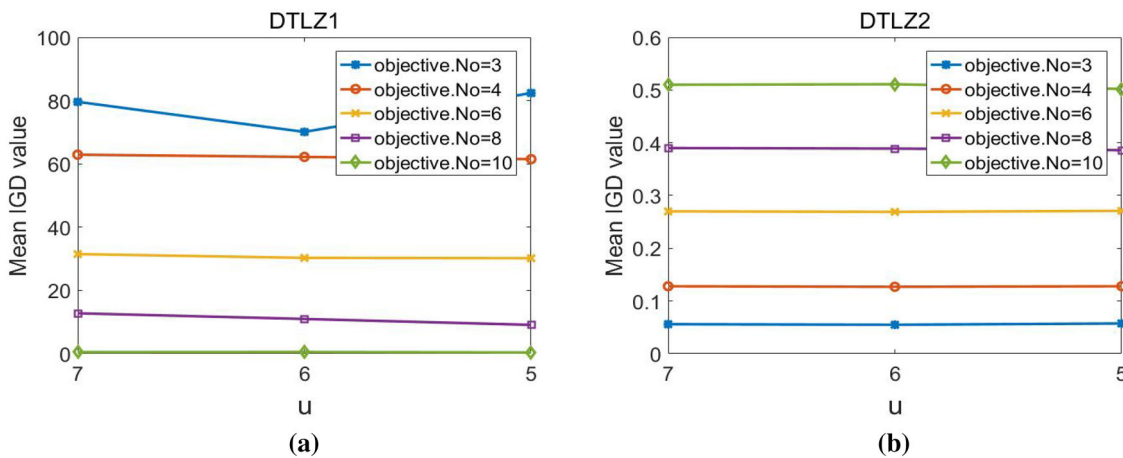


Fig. 6 The mean IGD values obtained by TSEMO with different threshold u

Furthermore, to verify the effectiveness of TSEMO on many-objective optimization problems, we conduct experiments on MaF1–MaF7, which are the modified versions of the DTLZ. And the results are presented in Table 3. For MOEAs without surrogates, TSEMO obtained 15 better results than RVEA and NSGA-II/SDR among 21 test problems, which further verified the effectiveness of adding the surrogate in MOEAs. We can see that our proposed TSEMO can obtain 11, 10 better results among 21 test problems than KRVEA and CSEA, and lose to win on 6 and 7 problems than KRVEA and CSEA, respectively, which shows that our proposed method is more efficient than both of these algorithms on MaF1–MaF7 test problems.

Parameter analysis

In the two-stage infill strategy, a total of u individuals are selected for expensive objective evaluations. For fairness, we set u to the same value of 5 as in the KRVEA [5]. In the first

stage of the infill strategy, K individuals are selected for real evaluations. And the remaining $u - K$ individuals are selected in the second stage. The number of individuals selected in the first stage will affect individuals with large uncertainty found in the second stage and ultimately affect finding a good Pareto front. Therefore, we first conducted experiments with $u = 5$, and different numbers of individuals selected in the first stage for expensive evaluations, i.e., $K = 5, 4, 3, 2, 1$. When $K = 5$, it means that individuals are selected for real evaluations only in the first stage. Figure 5 shows the mean IGD values of 20 independently runs obtained by TSEMO with the different numbers of individuals selected for exact evaluations in the first stage. From Fig. 5, we can see that when the number of individuals selected for exact evaluation in the first stage is set to 4, the performance of TSEMO is the best.

Then we further analyze the performance variation of the algorithm, when u , K increase further, and the number of infill samples in the second stage remains unchanged. Fig-

Table 4 Statistical results for IGD values obtained by TSEMO-I and TSEMO

Problem	<i>M</i>	TSEMO-I	TSEMO
DTLZ1	3	8.2589e+1 (1.84e+1)≈	8.2409e+1 (1.61e+1)
	4	6.1807e+1 (1.37e+1) ≈	6.4502e+1 (1.28e+1)
	6	2.9182e+1 (4.34e+0) ≈	3.0229e+1 (5.41e+0)
	8	1.2382e+1 (4.33e+0) –	9.1380e+0 (3.76e+0)
	10	5.6657e−1 (4.31e−1) ≈	4.0968e-1 (1.18e-1)
DTLZ2	3	5.6160e-2 (2.30e-3) ≈	5.7483e−2 (2.17e−3)
	4	1.3454e−1 (3.63e−3) –	1.2823e-1 (1.99e-3)
	6	2.8978e−1 (9.44e−3) –	2.7141e-1 (3.40e-3)
	8	4.3340e−1 (4.23e−2) –	3.8555e-1 (5.97e-3)
	10	5.7618e−1 (4.56e−2) –	5.0242e-1 (3.23e-2)
DTLZ3	3	1.8945e+2 (1.92e+1) ≈	2.1191e+2 (6.06e+1)
	4	1.8079e+2 (4.40e+1) ≈	1.7526e+2 (3.77e+1)
	6	9.6178e+1 (1.16e+1) ≈	1.0040e+2 (2.87e+1)
	8	4.0896e+1 (1.22e+1) –	3.2472e+1 (1.04e+1)
	10	1.6613e+0 (6.38e−1) ≈	1.5560e+0 (5.59e-1)
DTLZ4	3	4.0220e−1 (2.06e−1) ≈	3.1139e-1 (4.61e-2)
	4	4.3237e−1 (8.03e−2) ≈	3.9368e-1 (5.14e-2)
	6	5.0095e−1 (5.84e−2) ≈	4.9458e-1 (3.41e-2)
	8	5.4235e−1 (3.45e−2) ≈	5.2261e-1 (2.83e-2)
	10	5.7137e−1 (3.53e−2) ≈	5.6244e-1 (2.05e-2)
DTLZ5	3	2.3099e-2 (3.94e-3) ≈	2.3601e−2 (3.23e−3)
	4	2.7654e-2 (4.23e-3) ≈	3.1403e−2 (6.38e−3)
	6	2.5895e-2 (4.08e-3) ≈	2.8630e−2 (4.36e−3)
	8	1.9766e−2 (3.50e−3) ≈	1.9254e-2 (2.43e-3)
	10	1.2168e−2 (2.37e−3) –	1.0464e-2 (1.71e-3)
DTLZ6	3	1.8728e+0 (4.18e−1) ≈	1.7507e+0 (4.87e-1)
	4	1.2291e+0 (4.48e-1) ≈	1.4274e+0 (4.77e−1)
	6	1.0268e+0 (3.97e−1) ≈	8.7852e-1 (3.46e-1)
	8	4.3779e−1 (2.36e−1) ≈	3.6345e-1 (9.95e-2)
	10	4.2406e-2 (1.83e-2) ≈	4.6694e−2 (1.67e−2)
DTLZ7	3	1.4210e-1 (2.99e-2) ≈	1.4281e−1 (1.71e−2)
	4	2.3364e-1 (2.55e-2) ≈	2.4455e−1 (2.21e−2)
	6	5.0016e-1 (1.96e-2) ≈	5.0508e−1 (2.38e−2)
	8	8.1152e-1 (2.80e-2) ≈	8.1673e−1 (4.02e−2)
	10	1.1521e+0 (2.26e−2) ≈	1.1391e+0 (2.24e-2)
+ / − / ≈		0/7/28	

The best results are highlighted

ure 6 shows the mean IGD values of 20 independently runs obtained by TSEMO with the different numbers of *u* individuals selected for exact evaluations. From Fig. 6, we can see that there are no significant differences among the performance of TSEMO on 3, 4, 6, 8, 10 objectives of DTLZ2 when *u* increases further. On the low-dimensional 3 objectives of DTLZ1, the algorithm performs best when *u* = 6, *K* = 5. However, on the high-dimensional DTLZ1 of 4, 6, 8, 10 objectives, the algorithm performs best when *u* = 5, *K* = 4. Thus, we set *u* = 5 and *K* = 4 in our method

for solving the expensive many-objective optimization problems.

Effectiveness of the two-stage infill strategy

In this part, we investigate the effects of the two-stage infill strategy by comparing TSEMO with another variant TSEMO-I, which only selects individuals for exact evaluations in the first stage. Table 4 gives the statistical results of our proposed TSEMO, and TSEMO-I. From Table 4,

Table 5 Statistical results for IGD values obtained by TSEMO-II and TSEMO

Problem	M	TSEMO-II	TSEMO
DTLZ2	3	1.8534e−1 (3.76e−2) −	5.7483e-2 (2.17e-3)
	4	2.5978e−1 (4.11e−2) −	1.2823e-1 (1.99e-3)
	6	3.6333e−1 (3.73e−2) −	2.7141e-1 (3.40e-3)
	8	4.5395e−1 (2.11e−2) −	3.8555e-1 (5.97e-3)
	10	4.9455e-1 (1.55e-2) ≈	5.0242e−1 (3.23e−2)
DTLZ5	3	1.0441e−1 (4.09e−2) −	2.3601e-2 (3.23e-3)
	4	1.0950e−1 (3.15e−2) −	3.1403e-2 (6.38e-3)
	6	6.9213e−2 (1.97e−2) −	2.8630e-2 (4.36e-3)
	8	5.0860e−2 (1.53e−2) −	1.9254e-2 (2.43e-3)
	10	1.7743e−2 (4.08e−3) −	1.0464e-2 (1.71e-3)
DTLZ6	3	1.7087e+0 (3.21e-1) ≈	1.7507e+0 (4.87e−1)
	4	1.3937e+0 (2.56e-1) ≈	1.4274e+0 (4.77e−1)
	6	1.0651e+0 (2.12e−1) −	8.7852e-1 (3.46e-1)
	8	5.2791e−1 (1.68e−1) −	3.6345e-1 (9.95e-2)
	10	7.7568e−2 (2.39e−2) −	4.6694e-2 (1.67e-2)
+ / − / ≈		0/12/3	

The best results are highlighted

we can see that TSEMO obtained 7/35 better results than TSEMO-I. Compared to TSEMO-I, our TSEMO obtained 0 worse and 28 comparable results than TSEMO-I. TSEMO-I without the second stage sampling strategy has worse performance mainly on DTLZ1 with 8 objectives, DTLZ2 with 3, 4, 6, 8 and 10 objectives, DTLZ3 with 8 objectives, and DTLZ5 with 10 objectives. The reason we analyze is that DTLZ2 is used to investigate the diversity and distribution of the algorithm and the second stage sampling strategy has a potential benefit to improve the diversity of algorithms. For other problems, like DTLZ4 and DTLZ7, TSEMO-I showed comparable performance with TSEMO. Therefore, the effectiveness of the second stage infill sampling strategy cannot be ignored and the two-stage infill strategy is adopted to be used in our method.

Effectiveness of the two different kernel functions used in the surrogate ensemble

In this paper, we adopt two different kernel functions of RBF models to construct a surrogate-ensemble and utilize the ensemble to approximate each objective function. To see the efficiency of using surrogate-ensemble, we compare TSEMO with another variant TSEMO-II, where the surrogate-ensemble is replaced with GP, the uncertainty is provided by GP, and the two-stage infill strategy and the strategy of revising the population are employed. The average IGD results of TSEMO-II and TSEMO based on 20 independent runs are presented in Table 5. As can be seen in Table 5, TSEMO performed 12 better, 3 comparable, and 0 worse results among a total of 15 problems than TSEMO-II.

TSEMO performs well on most test problems. The results showed that TSEMO can achieve a better balance between convergence and diversity. This may be attributed to that the two different kernels of RBF can provide more accurate fitness predictions and the uncertainty information provided by our method is useful. Thus, we use the surrogate ensemble to approximate each objective function.

Effectiveness of revising the population

In this section, we investigate the effects of revising the population by comparing TSEMO with another variant TSEMO-R, which removes the strategy of revising the population. The average IGD results of TSEMO-R and TSEMO based on 20 independent runs on DTLZ1-DTLZ7 problems with 3, 4, 6, 8, and 10 objectives are presented in Table 6. As can be seen in Table 6, TSEMO performed 5 better, 25 comparable, and 0 worse results among a total of 35 problems than TSEMO-R. TSEMO performs better on DTLZ2 with 10 objectives, DTLZ5 with 8 and 10 objectives and DTLZ6 with 3 and 4 objectives. The results showed that the strategy of revising the population is effective for the surrogate-assisted multi-objective optimization problems.

Conclusion

In this paper, we propose a two-stage infill strategy and surrogate-ensemble assisted expensive many-objective optimization (TSEMO) method. To improve the robustness of the surrogate model, we use two kernel functions of RBF to con-

Table 6 Statistical results for IGD values obtained by TSEMO-R and TSEMO for DTLZ1-DTLZ7

Problem	M	TSEMO-R	TSEMO
DTLZ1	3	8.3799e+1 (1.60e+1) \approx	8.2409e+1 (1.61e+1)
	4	5.7835e+1 (1.43e+1) \approx	6.4502e+1 (1.28e+1)
	6	2.9962e+1 (4.55e+0) \approx	3.0229e+1 (5.41e+0)
	8	8.7329e+0 (3.07e+0) \approx	9.1380e+0 (3.76e+0)
	10	4.2367e−1 (1.52e−1) \approx	4.0968e-1 (1.18e-1)
DTLZ2	3	5.7477e-2 (2.56e-3) \approx	5.7483e−2 (2.17e−3)
	4	1.2822e-1 (3.20e-3) \approx	1.2823e−1 (1.99e−3)
	6	2.7217e−1 (5.50e−3) \approx	2.7141e-1 (3.40e-3)
	8	3.8813e−1 (1.23e−2) \approx	3.8555e-1 (5.97e-3)
	10	5.1958e−1 (2.19e−2) −	5.0242e-1 (3.23e-2)
DTLZ3	3	1.9159e+2 (2.77e+1) \approx	2.1191e+2 (6.06e+1)
	4	1.8175e+2 (4.86e+1) \approx	1.7526e+2 (3.77e+1)
	6	1.0924e+2 (1.96e+1) \approx	1.0040e+2 (2.87e+1)
	8	3.8663e+1 (1.45e+1) \approx	3.2472e+1 (1.04e+1)
	10	1.4728e+0 (4.58e-1) \approx	1.5560e+0 (5.59e−1)
DTLZ4	3	3.2031e−1 (6.70e−2) \approx	3.1139e-1 (4.61e-2)
	4	3.9581e−1 (3.37e−2) \approx	3.9368e-1 (5.14e-2)
	6	5.2076e−1 (4.80e−2) \approx	4.9458e-1 (3.41e-2)
	8	5.2465e−1 (3.12e−2) \approx	5.2261e-1 (2.83e-2)
	10	5.8204e−1 (3.30e−2) \approx	5.6244e-1 (2.05e-2)
DTLZ5	3	2.2435e-2 (3.17e-3) \approx	2.3601e−2 (3.23e−3)
	4	2.9782e-2 (5.65e-3) \approx	3.1403e−2 (6.38e−3)
	6	2.6321e-2 (4.91e-3) \approx	2.8630e−2 (4.36e−3)
	8	2.7590e−2 (3.24e−3) −	1.9254e-2 (2.43e-3)
	10	1.1568e−2 (1.64e−3) −	1.0464e-2 (1.71e-3)
DTLZ6	3	2.2876e+0 (6.61e−1) −	1.7507e+0 (4.87e-1)
	4	1.8655e+0 (5.76e−1) −	1.4274e+0 (4.77e-1)
	6	1.0711e+0 (3.94e−1) \approx	8.7852e-1 (3.46e-1)
	8	3.8225e−1 (1.53e−1) \approx	3.6345e-1 (9.95e-2)
	10	4.5238e-2 (1.45e-2) \approx	4.6694e−2 (1.67e−2)
DTLZ7	3	1.5304e−1 (2.58e−2) \approx	1.4281e-1 (1.71e-2)
	4	2.3883e-1 (2.87e-2) \approx	2.4455e−1 (2.21e−2)
	6	4.9962e-1 (2.24e-2) \approx	5.0508e−1 (2.38e−2)
	8	8.0918e-1 (2.50e-2) \approx	8.1673e−1 (4.02e−2)
	10	1.1460e+0 (2.42e−2) \approx	1.1391e+0 (2.24e-2)
+ / − / \approx		0/5/30	

The best results are highlighted

struct a surrogate ensemble. After optimization, a two-stage infill strategy is proposed to select individuals for expensive evaluations. The approximated variance of the surrogate ensemble and approximated fitness are considered adaptively in the first stage. And historical information of the approximated variance by the surrogate ensemble is considered in the second stage. To avoid the approximated errors from misleading the search direction, a strategy of revising the population is proposed.

We compare our method with three surrogate-assisted MOEAs and two non-surrogate-assisted MOEAs on two sets of test problems. The results show that TSEMO achieves better performance than the compared algorithms on the limited number of expensive function evaluations. In our work, the surrogate-ensemble, two-stage infill strategy, and strategy of revising the population are effective for solving expensive MOPs. However, for multimodal problems, like DTLZ3 and MaF4, TSEMO still needs more expensive function evaluations to find the true PFs. Therefore, a more effective

optimization process and infill strategy should be developed for solving these complex problems.

Declarations

Competing interests The authors declare that they have no conflict of interest.

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

- Bosman P, Thierens D (2003) The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Trans Evol Comput* 7(2):174–188
- Cai X, Gao L, Li X (2020) Efficient generalized surrogate-assisted evolutionary algorithm for high-dimensional expensive problems. *IEEE Trans Evol Comput* 24(2):365–379
- 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
- Cheng R, Rodemann T, Fischer M, Olhofer M, Jin Y (2017) Evolutionary many-objective optimization of hybrid electric vehicle control: from general optimization to preference articulation. *IEEE Trans Emerg Top Comput Intell* 1(2):97–111
- Chugh T, Jin Y, Miettinen K, Hakanen J, Sindhya K (2018) A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. *IEEE Trans Evol Comput* 22(1):129–142
- Deb K (2001) Multi-objective optimization using evolutionary algorithms, vol 16. Wiley, Chichester
- Deb K, Goyal M (1999) A combined genetic adaptive search (geneas) for engineering design. *Comput Sci Inform* 26:30–45
- Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Trans Evol Comput* 6(2):182–197
- Emmerich M, Giannakoglou K, Naujoks B (2006) Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodells. *IEEE Trans Evol Comput* 10(4):421–439
- Gu Q, Zhou Y, Li X, Ruan S (2021) A surrogate-assisted radial space division evolutionary algorithm for expensive many-objective optimization problems. *Appl Soft Comput* 111:107703
- Guo D, Wang X, Gao K, Jin Y, Ding J, Chai T (2022) Evolutionary optimization of high-dimensional multiobjective and many-objective expensive problems assisted by a dropout neural network. *IEEE Trans Syst Man Cybern Syst* 52(4):2084–2097
- Jia L, Wang Y, Fan L (2014) Multiobjective bilevel optimization for production-distribution planning problems using hybrid genetic algorithm. *Integrat Comput Aided Eng* 21(1):77–90
- Jin Y (2011) Surrogate-assisted evolutionary computation: recent advances and future challenges. *Swarm Evol Comput* 1(2):61–70
- Jin Y, Wang H, Chugh T, Guo D, Miettinen K (2019) Data-driven evolutionary optimization: an overview and case studies. *IEEE Trans Evol Comput* 23(3):442–458
- 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
- Li F, Shen W, Cai X, Gao L, Gary Wang G (2020) A fast surrogate-assisted particle swarm optimization algorithm for computationally expensive problems. *Appl Soft Comput* 92:106303
- Lin Q, Wu X, Ma L, Li J, Gong M, Coello CAC (2021) An ensemble surrogate-based framework for expensive multiobjective evolutionary optimization. *IEEE Trans Evol Comput* 20:1
- Liu B, Zhang Q, Gielen GGE (2014) A gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems. *IEEE Trans Evol Comput* 18(2):180–192
- Liu Y, Liu J, Jin Y (2021) Surrogate-assisted multipopulation particle swarm optimizer for high-dimensional expensive optimization. *IEEE Trans Syst Man Cybern Syst* 20:1–14
- Mckay MD, Beckman RJ, Conover WJ (1979) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 21(2):239–245
- Namura N, Shimoyama K, Obayashi S (2017) Expected improvement of penalty-based boundary intersection for expensive multi-objective optimization. *IEEE Trans Evol Comput* 21(6):898–913
- Pan JS, Liu N, Chu SC, Lai T (2021) An efficient surrogate-assisted hybrid optimization algorithm for expensive optimization problems. *Inf Sci* 561:304–325
- Pan L, He C, Tian Y, Wang H, Zhang X, Jin Y (2019) A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization. *IEEE Trans Evol Comput* 23(1):74–88
- Pires EJS, de Moura Oliveira PB, Machado JAT (2004) Multi-objective genetic manipulator trajectory planner. In: Raidl GR, Cagnoni S, Branke J, Corne DW, Drechsler R, Jin Y, Johnson CG, Machado P, Marchiori E, Rothlauf F, Smith GD, Squillero G (eds) *Appl Evol Comput*. Springer, Heidelberg, pp 219–229
- Ponweiser W, Wagner T, Vincze M (2008) Clustered multiple generalized expected improvement: A novel infill sampling criterion for surrogate models. In: 2008 IEEE congress on evolutionary computation (IEEE world congress on computational intelligence), pp 3515–3522
- Song Z, Wang H, He C, Jin Y (2021) A kriging-assisted two-archive evolutionary algorithm for expensive many-objective optimization. *IEEE Trans Evol Comput* 25(6):1013–1027
- Sun C, Jin Y, Zeng JC, Yu Y (2015) A two-layer surrogate-assisted particle swarm optimization algorithm. *Soft Comput* 19(6):1461–1475
- Sun C, Jin Y, Cheng R, Ding J, Zeng J (2017) Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems. *IEEE Trans Evol Comput* 21(4):644–660
- Tian J, Tan Y, Zeng J, Sun C, Jin Y (2019) Multiobjective infill criterion driven gaussian process-assisted particle swarm optimization of high-dimensional expensive problems. *IEEE Trans Evol Comput* 23(3):459–472
- Tian Y, Cheng R, Zhang X, Cheng F, Jin Y (2018) An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility. *IEEE Trans Evol Comput* 22(4):609–622
- Tian Y, Cheng R, Zhang X, Su Y, Jin Y (2019) A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization. *IEEE Trans Evol Comput* 23(2):331–345

32. Wang H, Jin Y, Doherty J (2017) Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems. *IEEE Trans Cybern* 47(9):2664–2677
33. Wang H, Jin Y, Sun C, Doherty J (2019) Offline data-driven evolutionary optimization using selective surrogate ensembles. *IEEE Trans Evol Comput* 23(2):203–216
34. Wang X, Jin Y, Schmitt S, Olhofer M (2020) An adaptive Bayesian approach to surrogate-assisted evolutionary multi-objective optimization. *Inf Sci* 519:317–331
35. Yang C, Ding J, Jin Y, Chai T (2020) Offline data-driven multiobjective optimization: knowledge transfer between surrogates and generation of final solutions. *IEEE Trans Evol Comput* 24(3):409–423
36. Yu H, Tan Y, Zeng J, Sun C, Jin Y (2018) Surrogate-assisted hierarchical particle swarm optimization. *Inf Sci* 454–455:59–72
37. Yu H, Kang L, Tan Y, Zeng JC, Sun C (2021) A multi-model assisted differential evolution algorithm for computationally expensive optimization problems. *Complex Intell Syst* 7:2347–2371
38. Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans Evol Comput* 11(6):712–731
39. Zhang Q, Liu W, Tsang E, Virginas B (2010) Expensive multiobjective optimization by MOEA/D with Gaussian Process model. *IEEE Trans Evol Comput* 14(3):456–474
40. Zhao Y, Sun C, Zeng J, Tan Y, Zhang G (2021) A surrogate-ensemble assisted expensive many-objective optimization. *Knowl Based Syst* 211:106520
41. Zhao Y, Zeng J, Tan Y (2021) Neighborhood samples and surrogate assisted multi-objective evolutionary algorithm for expensive many-objective optimization problems. *Appl Soft Comput* 105:107268
42. Zitzler E, Künzli S (2004) Indicator-based selection in multiobjective search. *Parallel problem solving from nature*. Springer, Berlin, pp 832–842

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