Information Sciences 666 (2024) 120405

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Surrogate-assisted PSO with archive-based neighborhood search for medium-dimensional expensive multi-objective problems

Mingyuan Yu [a,c], Zhou Wu [b,*], Jing Liang [c], Caitong Yue [a,c]

[a] School of Electrical and Information Engineering, Zhengzhou University, Zhengzhou 450001, PR China
[b] School of Automation, Chongqing University, Chongqing 40000, PR China
[c] State Key Laboratory of Intelligent Agricultural Power Equipment, Zhengzhou University, Zhengzhou, 450001, PR China

ARTICLE INFO

Keywords:
Expensive multi-objective problem (MOP)
Surrogate-assisted evolutionary algorithm (SAEA)
Particle swarm optimization (PSO)
Neighborhood search
Infill criterion

ABSTRACT

Thousands of real function evaluations are not burdensome when a surrogate-assisted evolutionary algorithm (SAEA) is used to solve expensive multi-objective optimization problems (MOPs). To reduce the computational overhead, this paper studies a surrogate-assisted multi-objective particle swarm optimization algorithm, named SaMOPSO_NS, in which an external archive-based neighborhood search, as a local search, and a *pbest*-dominance-based infill criterion are newly developed. The local search works hard to refine exploitation around the current non-dominated individuals once the trigger mechanism is activated, while the *pbest*-dominance-based infill criterion chooses non-dominated individuals predicted by an ensemble surrogate for actual evaluations. With the collaborative efforts between the local search strategy and the infill criterion, computing resources are more efficiently allocated. Three types of benchmark test instances with different dimensions as well as an engineering expensive MOP are used to examine the proposed algorithm. Experimental results demonstrate that the proposed algorithm significantly outperforms its rivals with fewer real evaluations on most medium-dimensional MOPs. Moreover, the optimized electromagnetic acoustic transducers achieved an amplitude and amplitude ratio of 9.666E-07 mm and 0.1328 respectively, markedly outpacing previously reported results.

1. Introduction

Multi-objective evolutionary algorithms (MOEAs) often require a significant number of function evaluations (FEs) to obtain a set of high-quality non-dominant solutions, commonly known as Pareto optimal solutions (PS) in the decision space. Nevertheless, MOEAs can often fall short in instances where function evaluation turns exceedingly expensive, particularly in real-world multi-objective optimization problems (MOPs) such as structural optimization [1], antenna design and optimization [2], and trauma systems optimization [3], among others. These problems typically involve a multitude of time-intensive simulations or computationally demanding calculations, which is why they are categorically referred to as expensive MOPs [4,5].

In order to decrease the number of expensive FEs, the surrogate model, also known as the metamodel or approximation model, is often employed to replace the expensive evaluation or simulation because it can fit the input-output relationship of a function but is cheap to evaluate. Prominent and widely-used surrogate models include the Gaussian process (GP) [6], polynomial regression surface (PRS) [7], radial basis function (RBF) network [8,9], and support vector machines (SVM) [3]. Other machine learning models that can

* Corresponding author.
E-mail address: zhouwu@cqu.edu.cn (Z. Wu).

https://doi.org/10.1016/j.ins.2024.120405
Received 6 July 2022; Received in revised form 26 February 2024; Accepted 28 February 2024
Available online 2 March 2024

be applied as surrogate models are regression or classification techniques [10]. However, given the pros and cons between different models, employing multi-surrogate reorganizations [11,12], such as ensemble surrogate, and hybrid surrogate, have proven to exhibit more robustness compared to a single surrogate.

Thus far, a significant body of literature has emerged on the topic of using surrogate-assisted evolutionary algorithms (SAEAs) to tackle expensive multi-objective optimization problems (MOPs) [13] and even many-objective optimization problems (MaOPs). In an effort to improve the convergence of the optimization process, Yang et al. [14] leveraged the knowledge transfer method to transmit insights from the coarse surrogate to the fine surrogate. Additionally, they proposed a reference vector-based (RV) strategy for generating the final solution sets, aiming to minimize errors produced by the surrogate. Wang et al. [15] fine-tuned the hyper-parameters within the acquisition function based on search dynamics. This allowed for discerning which candidate solutions merited actual evaluation. They further proposed a sampling selection criterion switching strategy to achieve equilibrium between exploration and exploitation. In cases of expensive MOPs with varying evaluation times, Wang et al. [16] devised a transfer learning scheme within the SAEA framework that facilitated knowledge transfer between two diverse objective functions. Chen et al. [17] implemented a cluster-based neighborhood regression model as a surrogate to support the MOEA/D algorithm in processing expensive MOPs. To solve expensive MOPs with medium dimensions, a two-infill criteria-driven SAEA was suggested [18]. Here, the convergence-based and diversity-based criteria were jointly employed to select potentially successful individuals for precise evaluations. High-dimensional expensive MOPs presented an even greater challenge. To this end, Xiang et al. [19] formulated a clustering-based SAEA and applied it to resolve the real-world shelter location problem. They recommended a clustering strategy to transform the surrogate of the high-dimensional problem into a low-dimensional one. Guo et al. [20] countered the potential computational complexity of the Gaussian process (GPs), which tends to increase with dimensionality. They developed a heterogeneous ensemble, capable of estimating the uncertainty, as an alternative to GPs. Lv et al. [21] offered the Pareto active learning-based surrogate-assisted particle swarm optimization (PAL-SAPSO) algorithm to manage expensive MOPs. They employed a classification strategy in which the improved ε-PAL technique is used to identify candidate solutions categorized as Pareto optimal and not Pareto-optimal. To boost computational efficiency, Wang et al. [5] created an adaptive sub-problem selection strategy which could pinpoint and select the most promising sub-problems for additional surrogate-assisted optimization.

Recently, in an effort to adeptly solve expensive dynamic MOPs, Liu et al. [22] incorporated both a training data augmentation strategy and a sampling pool expansion strategy into the Kriging-assisted RV-guided EA (KRVEA). To improve expensive MaOPs-solving, to this end, the KRVEA was designed to handle the expensive MaOPs with more than three objectives [23]. Jiang et al. [24] designed a clustering-assisted multi-objective infill criterion to improve the accuracy of the GP model. Additionally, they also proposed an efficient constrained global optimization algorithm to tackle situations where both the objective function and constraints are expensive in the context of MOPs. Song et al. [10] introduced a Kriging-assisted Two_Arch2 (KTA2) algorithm, which incorporates two archives, namely the convergence archive and the diversity archive, dedicated to improving convergence and diversity aspects, respectively. Taking a different approach, Pan et al. [25] employed an artificial neural network (ANN) as a surrogate to predict the dominant relationship between the candidate solutions and the selected reference solutions, and experimental results demonstrate that the proposed CSEA outperforms the state-of-the-art SAEAs for expensive MaOPs.

Despite in principle, the aforementioned SAEAs are indeed capable of solving the expensive MOPs, plenty of real FEs are consumed to obtain an acceptable solution. As an illustration, KMOPSO [26] consumes approximately 5000 and 15,000 real FEs to handle MOPs with 10 and 30 decision variables, respectively. The PAL-SAPSO algorithm [21] requires 2000 real FEs for various multi-objective benchmark problems with 10 decision variables, while the Heterogeneous Ensemble-based MOEA (HeE-MOEA) [21] needs 1000 real FEs for a MOP with 80 decision variables. It's crucial to highlight that so many real FEs to be consumed is impractical especially for those extremely expensive MOPs, even though the number has been reduced a lot compared to those used in the MOEAs literature without a surrogate. To compensate for the shortcomings of existing methods, it's essential to devise a suitable SAEA that can efficiently derive acceptable non-dominated solutions with minimal computational overhead, primarily in the range of hundreds of real FEs.

Inspired by this challenge, our goal is to solve medium-dimensional expensive MOPs using only hundreds of real FEs. To accomplish this, we proposed a surrogate-assisted MOPSO coupled with an archive-based neighborhood search (NS) algorithm, henceforth referred to as SaMOPSO_NS. In the proposed algorithm, the MOPSO algorithm [21] was used as a canonical optimizer taking into account its fast convergence and simple implementation. Meanwhile, the *pbest*-dominated infill criterion and the external archive-based neighborhood search (EXA-NS) work together to strike a balance between diversity and convergence. The proposed infill criterion works hard to select the predicted individuals who can dominate their previous best ones for expensive FEs, so as to explore the individual well-distributed areas. While EXA-NS strategy, as a local search, strives to exploit the landscape around the current non-dominated set, where individuals are likely to be close to the true PF. In addition, It is crucial to underscore that the primary intention of this paper is to provide innovative perspectives on handling computationally expensive MOPs, not to defeat existing advanced SAEAs. The principal contributions of this paper can be summarized as follows.

- An enhanced MOPSO algorithm is studied, in which the neighborhood search, acting as a local search, is specifically applied through a trigger mechanism for updating the external archive of the population.
- To enhance the reliability of the surrogate model, a newly developed *pbest*-dominant-based infill criterion and an ensemble surrogate are utilized. The ensemble surrogate comprises multiple RBF models with distinct input features, resulting in ensemble members that are both highly diverse and sufficiently accurate.
- The SaMOPSO_NS algorithm is designed to solve medium-dimensional expensive MOPs with hundreds of real FEs. Interestingly, the proposed algorithm can achieve better or more competitive experimental results than these compared SAEAs.

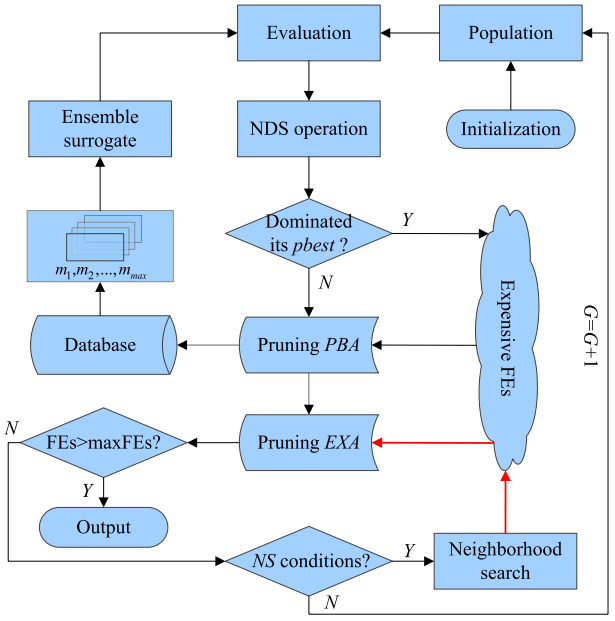


Fig. 1. The framework of the proposed SaMOPSO_NS algorithm.

The rest of this paper is organized as follows. Section II provides some preparatory knowledge related to the proposed algorithm. The main components of SaMOPSO_NS are then described in detail in Section III. Section IV presents and discusses the comprehensive experimental results. This paper is summarized in Section V.

2. Preparatory knowledge

In this section, we introduce the key knowledge about the MOPSO algorithm and neighborhood search.

2.1. Multi-objective PSO algorithm (MOPSO)

In PSO, each particle learns from its previous best individual, denoted as *pbest*, and the global best individual, denoted as *gbest*, and updates the particle search state by communicating with both of them. The PSO starts with an initial population randomly generated in the search space and ends when the termination condition is met. In each iteration, the particle within-population changes its position and velocity as follows

$$v_{i,G} \leftarrow \omega v_{i,G} + c_1 r_1 (pbest_i - x_{i,G}) + c_2 r_2 (gbest - x_{i,G}) \tag{2}$$

$$x_{i,G} \leftarrow x_{i,G} + v_{i,G} \tag{3}$$

where $v_{i,G}$ and $x_{i,G}$ respectively represent the velocity and position of the ith particle at the Gth iteration. $pbest_i$ is the position of the best fitness found so far for the ith particle. c_1 and c_2 are the acceleration factors that pull each particle to learn towards *pbest* and *gbest*, respectively. r_1 and r_2 are two random numbers independently generated within [0,1]. ω is the inertia weight used to make a balance between exploration and exploitation.

As high-speed convergence and easy implementation, the PSO algorithm has been widely developed to handle MOPs, that is, multi-objective particle swarm optimization algorithm [27]. The remarkable success of MOPSO can be attributed to the use of an external archive set. For most MOEAs, the external archive (denoted as EXA) is mainly used to store non-dominated solutions and guide particle updates to avoid premature convergence. In addition, the maintenance of individuals' previous best archives (denoted as PBA) improves the diversity of the population.

2.2. Neighborhood search

The main idea of neighborhood search is the individuals within the population will be attracted by their superior neighbor and repulsed by their inferior neighbor [28,29], and its mutation operation can be expressed as

$$v_{i,G} \leftarrow x_{i,G} + F_i \cdot (xc_{i,G} - x_{i,G}) + F_i \cdot (x_{i,G} - xw_{i,G}) \tag{6}$$

where the parameter F_i is the ith mutation factor which usually ranges in (0, 1]. In Eq. (6), the $xc_{i,G}$ and $xw_{i,G}$ are the superior neighbor

and inferior neighbor of $x_{i,G}$, respectively, and they can be identified as follows:

$$\begin{cases} xc_{i,G} = \text{argmin}_{f(x_{m,G})<f(x_{i,G})} \left\| x_{m,G} - x_{i,G} \right\| \\ xw_{i,G} = \text{argmin}_{f(x_{m,G})>f(x_{i,G})} \left\| x_{m,G} - x_{i,G} \right\| \end{cases}, \quad i, m = 1, 2, ..., NP \tag{7}$$

where $f(\bullet)$ is the objective function and we here assume that it is a single-objective optimization problem. $||x_{m,G} - x_{i,G}||$ is the Euclidean distance between $x_{m,G}$ and $x_{i,G}$ at the Gth iteration. NP is the size of the population. Other parameters are the same as those in the aforementioned MOPSO. If no individual is better than $x_{i,G}$ in the whole population, then we define $x_{i,G}$ as its superior neighbor. If $x_{i,G}$ is the worst individual in the whole population, then itself is defined as its inferior neighbor. For MOPs, it is noted that the superior neighbor and inferior neighbor are identified by the non-dominant relationship [30].

3. Proposed approach

3.1. Proposed SaMOPSO_NS algorithm

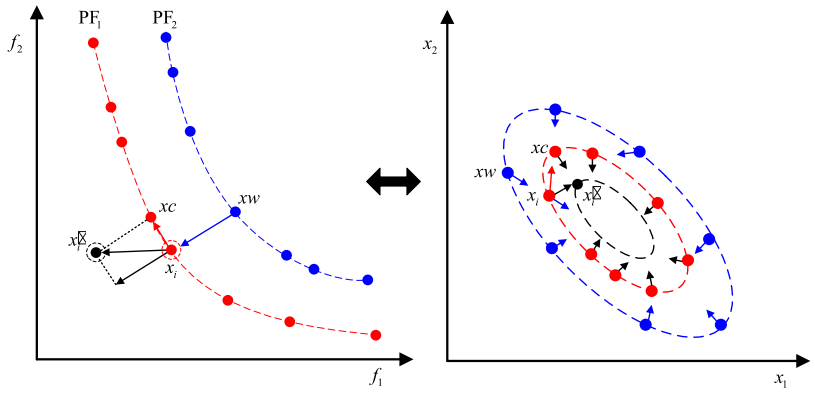
In this paper, we propose a surrogate-assisted PSO algorithm, integrated with an external archive-based neighborhood search algorithm, specifically designed for solving computationally expensive MOPs. A comprehensive outline of the SaMOPSO_NS framework is depicted in Fig. 1 and the associated procedures detailed in Algorithm 1. The proposed algorithm distinguishes itself from the other surrogate-assisted MOEAs primarily due to the incorporation of a *pbest*-dominated infill criterion and the external archive-based neighborhood search strategy. These two operations are tailor-made for the MOPSO algorithm and do not apply to other MOEAs without an external archive. SaMOPSO_NS consists of three crucial modules, each of which will be extensively elucidated in the subsequent discussion.

During the initialization, the Latin hypercube sampling (LHS) method [31] is adopted to generate a certain number of initial sample points in the decision space, and these samples are evaluated by the expensive objective function. Every evaluated sample is saved in the database (DB). An initial population is then created by randomly selecting NP samples from the DB. Carry out a non-dominated sorting (NDS) [32] operation on the population to yield the initial PBA and EXA_1. During the ensemble generation phase, each objective function constructs its own ensemble surrogate. With a limited computational budget, the population is iteratively updated according to the evolutionary mechanism of domination-based MOEAs. The ensemble surrogate predicts the fitness values of all individuals within the population. Subsequently, the *pbest*-dominated infill criterion is implemented. This criterion precisely evaluates those predicted individuals who are not dominated by their own historical best individuals. These evaluated individuals are then stored in their own PBA, after which the PBA is pruned. Next, NDS is performed on all individuals within the PBA. If the two conditions are met at the same time, the neighborhood search is implemented. During neighborhood search operation, two external archives, EXA_1 and EXA_2, are used to store the individuals located in the current first PF (PF_1) and second PF (PF_2) respectively. Here, it should be emphasized that only all individuals in EXA_1 are subjected to a local search and the expensive FEs are carried out on their offspring. After the above steps, the EXA_1 is pruned and the EXA_2 is set to empty. Finally, the non-dominated solution set is output from EXA_1 when the maximum number of the expensive FEs is reached. From the entire process, we can see that the expensive FEs are mainly consumed in the proposed two key operations, which may be the main factor in the computational complexity of the proposed algorithm. In the following, we will delineate comprehensive specifics of the proposed *pbest*-dominated infill criterion and EXA-based neighborhood search.

3.2. pbest-dominated infill criterion

A surrogate model built is often of low accuracy in the initial stage, and as evolution goes on, more real samples should be pregnant to enhance their accuracy. Yet, the development of a real sample could be time-consuming. If too many unimportant individuals are being evaluated expensively, it may make little contribution to reducing the overall FEs. Thus, selecting individuals for precise evaluation judiciously is crucial. While some surrogate-assisted pre-screening criteria have been established, this paper contemplates a *pbest*-dominated-based individual selection strategy, for two primary justifications. Firstly, the particle evolution is primarily influenced by the *gbest* individual and its *pbest* individual in the PSO algorithm. The low-fidelity or inexact historical optima are bound to lead particle astray. To avoid this situation, we must ensure that the historical optima are reliable by evaluating them exactly. Secondly, employing the proposed criterion is conducive to improving both the accuracy and diversity of surrogates. If the predicted non-dominated individuals are real non-dominated ones, their precise evaluation aids in bolstering surrogate accuracy. On the contrary, if the predicted non-dominated individuals are not real ones, their precise assessment subsequently fosters an increase in PBA diversity. Given that the training data is extracted from the PBA, the extent of surrogate diversity also consequently witnesses improvement.

In the Gth generation, the ensemble surrogate is first taken to predict each individual's fitness values. If any individual belongs to PBA_i dominated by the ith predicted individual, the predicted ith individual will be exactly evaluated. The evaluated ith individual is then stored in its corresponding PBA_i. It should be pointed out that the predicted dominated individuals are not exactly evaluated in this paper. This is because such individuals might only have a slight probability of self-improvement and are therefore considered non-essential. The design of the infill criterion intends to minimize these non-essential FEs, allocating more FEs to the potential individuals for progressive evolution towards the true Pareto front. During the PBA pruning operation, the Crowding Distance (CD) mechanism [26] is applied to manage the archive size of each individual, with individuals exhibiting low CD values being eliminated. Lower CD values result in reduced population diversity. The meticulous process of executing this criterion is outlined in lines 11–23 of Algorithm



(a) Objective space (b) Decision space

Fig. 2. The illustration of the EXA-NS strategy.

1.

3.3. Archive-based neighborhood search

For expensive MOPs, one major task of surrogate-assisted MOPSO is to find a set of non-dominated individuals that are very close to or even cover the real PF within the limited FEs. For this purpose, the idea of neighborhood search is borrowed [29], which encourages the non-dominated individuals in the current obtained EXA to exploit their near potential area. Meanwhile, to ensure individuals are distributed as uniformly as possible along with the true PF, the CD mechanism is applied again to calculate their elimination probability. The proposed archive-based NS strategy is visually exemplified in Fig. 2.

In Fig. 2, the red and blue points represent the individuals on PF_1 and PF_2 at Gth generation, respectively. Given that each individual on PF_2 is dominated by all the individuals on PF_1, we consequently believe that all individuals on PF_1 are superior to those on PF_2. For each individual on PF_1, we define the individuals closest to it on PF_1 and PF_2 as its superior neighbor and inferior neighbor respectively. It can be observed from Fig. 2 that the individual x_i ($i = 1,\ldots, n\_EXA$) is attracted by its superior neighbor xc, and repelled by its inferior neighbor xw, where n_EXA is the size of EXA. As a result, a new individual, denoted as $x_i^{'}$, is engendered and driven by the two neighbors, as the black point shows. The new individual $x_i^{'}$ is exactly evaluated and then stored in EXA. The size of EXA will be pruned in the same way as that of PBA once it overflows. Herein, if the inverse generation distance (IGD) [33] metric of all individuals in EXA does not change within a certain generation interval, a neighborhood search will be triggered. This essentially signifies that the generation interval correlates with the frequency of a local search-a smaller interval means more frequent searches and vice versa. If EXA-NS is executed too many times, it may be detrimental to individual diversity because available FEs becomes less for the proposed criterion. Besides, we also cannot be sure that every NS is effectively implemented every time. On the other hand, the EXA-NS strategy might not contribute to accelerating algorithm convergence if the times are too fewer, which deviates from its original purpose. Therefore, a proper interval parameter will be determined in the subsequent experimental analysis. The execution procedure of the EXA-NS strategy is outlined from line 24 to 43 in Algorithm 1.

3.4. Ensemble generation

In this paper, the process of ensemble generation, which closely mirrors the previously reported selective bagging method [34], involves data extraction, model training, and model combination. Bagging (short for bootstrap aggregating) sampling [35], a method of data manipulation [36], is used to create varying inputs and enhance ensemble diversity. More explicitly, bagging sampling is independently performed m ($m \leq m_{max}$) times for each objective function, extracting m distnct data subsets from DB. Each data subset thus contains a randomized portion of the database which then serves as input to the model. Subsequently, m different models are trained (denoted as M_1, M_2,…, M_m), and next all the trained models are combined using a weight-sum formulation [37] to create an ensemble model. Furthermore, after initialization, the database is updated with every generation, as seen in Fig. 1. It is to randomly select one from each PBA_i and put it into the DB so that the DB size constantly equals NP. The specifics of this process can be seen in lines 8–10 of Algorithm 1.

An ensemble surrogate naturally supersedes a single one in regards to accuracy and robustness [38]. Ensembles built on diverse inputs tend to outperform those based on homogeneous inputs. The size of the ensemble members is vital as it dictates the number of models needed for combination, and to some degree, it influences prediction accuracy, ensemble diversity, and computational efficiency. If the membership size is too small, the ensemble starts to resemble a single surrogate, potentially compromising its diversity and robustness. Conversely, an overly large membership size, such as the 2000 RBF members recommended in [34], may impair computational efficiency due to time-intensive modeling. Besides, the performance of the ensemble model doesn't necessarily improve once the member size reaches a certain threshold. In our experimental study, the maximum membership size was set to m_{max}, hoping to

achieve sufficiently good prediction accuracy at relatively low computational cost. What also deserves expecting is the parameter m_{max} is discussed and analyzed in the experiment sections.

Algorithm 1. Pseudo-code of the proposed SaMOPSO_NS algorithm.

Algorithm 1: Pseudo-code of the proposed SaMOPSO_NS algorithm

1. **Initialization:** Set the number of maximum expensive *FEs* (*maxFEs*), Set the number of maximum generations (*maxG*), initialize the values of parameters such as the population size (*NP*), the maximum size of *EXA* and *PBA* (*n_EXA* and *n_PBA*), the size of ensemble membership (m_{max}), the local search interval (*NS_int*), and other necessary parameters for the algorithm to run.
2. Set EXA_1, PBA_i (i=1, 2,…, *NP*), and database (*DB*) to be the empty set.
3. Generation *NP* individuals using LHS and evaluate them exactly.
4. Store all the evaluated individuals into *BD*, and generate PBA_i .
5. Select non-dominated solutions using NDS in the population, and store them in EXA_1.
6. **while** *FEs*<*maxFEs* **do**
7. **while** *G*<*maxG* **do**

---------- *** Ensemble Generation *** ----------

8. Randomly select ones from each PBA_i (i=1, 2,…, *NP*), and put it into *DB*.
9. Select *NP*/2 samples from *DB* to build an RBF model, and repeat this process m_{max} times.
10. Build an ensemble surrogate consisting of m_{max} RBF models for each objective function.

---------- *** pbest-Dominated Infill Criterion *** ----------

11. **for** i←1: *NP* **do**
12. Update individual based on Step4 (a)-(b) of MOPSO in Section II.
13. Evaluate the individual x_i using the ensemble model.
14. **if** $\{x_i \mid x_i \notin PBA_i\} \prec \{\exists\ pbest \mid pbest \in PBA_i\}$
15. Perform the expensive *FEs* on x_i
16. *FEs*←*FEs*+1
17. Put the evaluated x_i into PBA_i.
18. Update PBA_i using NDS and check the size of PBA_i.
19. **if** $|PBA_i| > n\_PBA_i$
20. Calculate CD of all individuals within-PBA_i and delete the extra ones with small CD values.
21. **end if** (line 19)
22. **end if** (line 14)
23. **end for** (line 11)

---------- ***EXA-based Neighborhood Search*** ----------

24. Put all non-dominated individuals from *PBA* to EXA_1.
25. Update EXA_1 using NDS and check its size.
26. **if** $|EXA_1| > n\_EXA$
27. Calculate the CD of all individuals within-EXA_1 and delete the extra ones with small CD values.
28. **end if** (line 26)
29. Calculate IGD at Gth iteration: IGD_G
30. **if** $G > NS\_int$ & $IGD_G - IGD_{G\text{-}NS\_int} = 0$
31. Perform NDS on *PBA*, and then put the individuals in PF_2 to EXA_2
32. Dis1←distance($x_{1,i} \in EXA_1$, $x_{1,j} \in EXA_1$), $i \neq j$=1,…,$|EXA_1|$.
33. Dis2←distance($x_{1,i} \in EXA_1$, $x_{2,j} \in EXA_2$), $i \neq j$=1,…,$|EXA_2|$.
34. **for** i←1: $|EXA_1|$ **do**
35. Find the individual closest to $x_{i,EXA1}$ from Dis1, and then define it as the superior neighbor of $x_{i,EXA1}$.
36. Find the individual closest to $x_{i,EXA1}$ from Dis2, and then define it as the inferior neighbor of $x_{i,EXA1}$.
37. Update $x_{i,EXA1}$ according to Eq. (3) and Eq. (2).
38. Perform the expensive *FEs* on the updated individual.
39. *FEs*←*FEs*+1
40. Store the exactly evaluated individual in EXA_1.
41. **end for** (line 34)
42. Set EXA_2 to empty and repeat line25-line29.
43. **end if** (line 30)
44. Iteration count: G←G+1
45. **if** *FEs*>*maxFEs*
46. Break
47. **end while** (line 7)
48. **end while** (line 6)
49. **end if** (line 45)

Output: The non-dominated solution set

Note: 'A $\prec$ B' means A dominates B in line 14.

4. Simulation results and discussion

In this section, we initially analyze the impacts of certain crucial parameters on the performance of our proposed SaMOPSO_NS algorithm. As two variations of the SaMOPSO_NS, SaMOPSO and MOPSO_NS, are devised to examine the proposed algorithm's behavior. SaMOPSO is the same as SaMOPSO_NS only except that it does not use the EXA-NS strategy, while MOPSO_NS lacks the proposed infill criterion and surrogate. We compare SaMOPSO_NS with variants to discern whether the proposed *pbest*-dominated infill criterion or EXA-NS strategy enhances the original algorithm. Finally, we validate the proposed algorithm by comparing it with a few

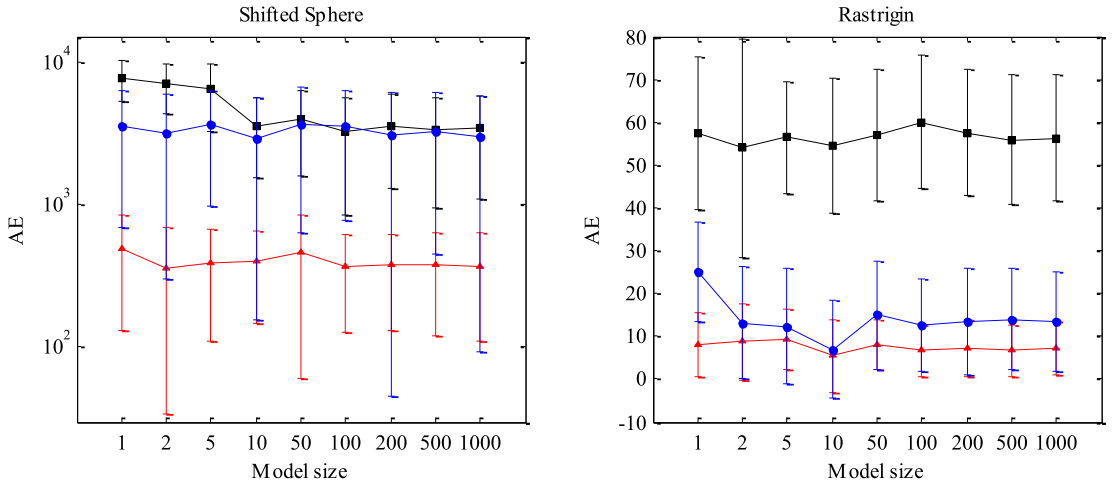


Fig. 3. The AE of different membership sizes on two functions with different dimensions.

state-of-the-art surrogate-assisted MOEAs including CSEA [25], ParEGO [39], MOEA/D-EGO [40], KRVEA [23], EIM-EGO [4] on UF [41], ZDT, DTLZ test instances [18] as well as an expensive multi-objective electromagnetic acoustic transducers (EMAT) optimization problem. All these test instances, with various characteristics, are minimization problems. Noted that all the compared algorithms are implemented in the PlatEMO platform [42], and their characteristics are summarized in [10]. For each test instance, two objectives are consistently maintained while the number of decision variables is configured to 10, 20, 30, and 50 respectively.

4.1. Experimental settings

1) For the proposed SaMOPSO_NS, specific parameters are set as below.

- The proposed algorithm uses the MOPSO as an optimizer to evolve the population and its parameter settings as recommended in the original paper [27].
- Parameter settings of the EXA-NS strategy refer to [29].
- The sizes of EXA and PBA are set to $n\_EXA = NP$, $n\_PBA = 10$.
- The population size is set to $NP = 50$. The size of the database is set to NP, and the number of training data is set to $NP/2$.
- The size of the ensemble membership is set to $m_{max} = 10$; the 'OWS_{full}' criterion [8] is adopted to build an ensemble surrogate. The generation interval of the neighborhood search is set to $NS\_int = 2$. The effects of m_{max} and NS_int on the proposed algorithm have been investigated in Section IV.B.
- The number of the maximum generations can be set to any integer no less than $maxFEs$. In an extreme case, if only one individual is accurately evaluated in each generation, then the $maxG = maxFEs$. Thus, $maxG$ is set $maxFEs$ here.
- The RBF interpolation model with the MQ function is used for the ensemble as the basis function shows good performance in other SAEAs [8].

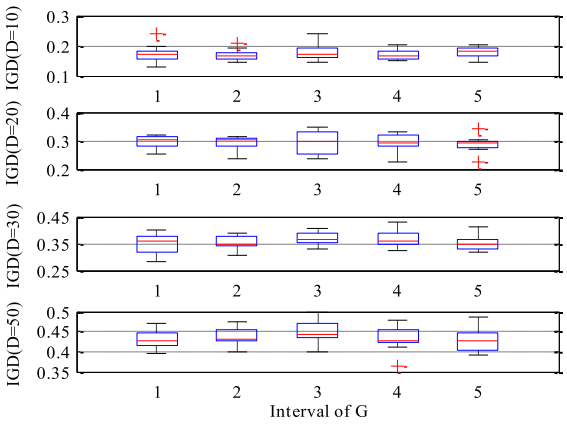
2) For the compared algorithms, their settings are as below.

- The population size is set to $NP = 100$ in this experiment.
- All algorithms run in parallel on the PlatEMO platform, and the recommended parameters settings as their original literature are adopted.
- The reference points of all test instances and the number of reference points are obtained from the PlatEMO platform.

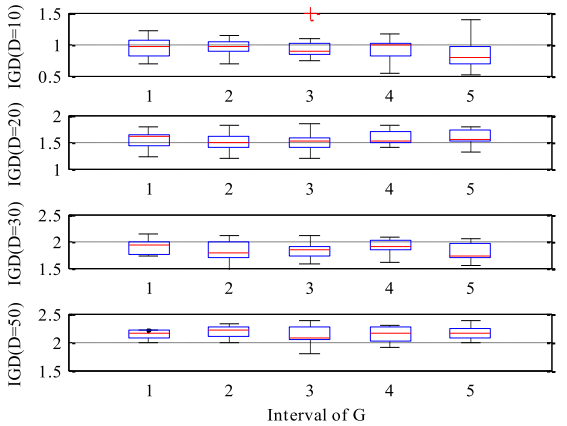
Here, all the algorithms performed 21 independent runs for each test instance with $D = 10$, 20, 30, and 50, where D is the dimension of decision variables. The predefined maximum number of the expensive FEs is set to $maxFEs = 300$, and the $maxFEs$ are exhausted as the stopping condition of an algorithm. IGD, as a commonly-used performance metrics indicator, is adopted to assess the performance of all compared algorithms by calculating the distance between the captured non-dominated individuals and the reference points. The smaller the IGD value, the better the solution set is. In our experiment, 10,000 reference points uniformly distributed along the PF are sampled for IGD calculation for each instance. All algorithms are carried out on Matlab R2014a and run on a PC with an Intel i7-2670QM CPU@2.50 GHz and 8 GB RAM.

4.2. Parameter analysis

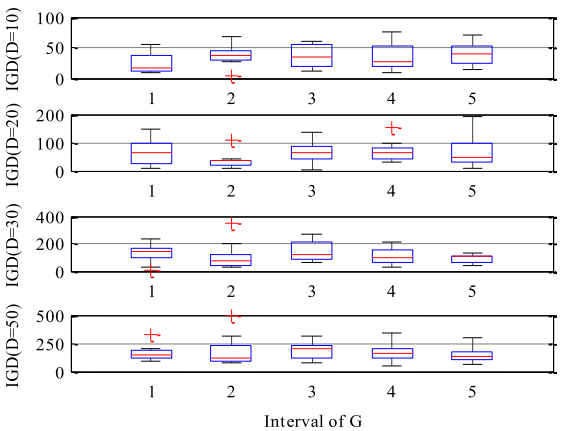
SaMOPSO_NS contains two critical parameters, m_{max} and NS_int, which as previously noted, are crucial for the ensemble model's



(a) UF2



(b) ZDT1



(c) DTLZ1

Fig. 4. IGD of different generation intervals on three test instances.

reliability and the efficient allocation of computing resources. The performance of SaMOPSO_NS algorithm can be influenced by varying values of m_{max} and *NS_int*. To ensure a broad applicability, we initially explore the impact of m_{max} through the lens of the unimodal Sphere benchmark function and the multimodal Rastrigin benchmark function [37], both exemplifying standard single-objective optimization problems. Although we intend to address expensive MOPs, this does not affect the sensitive analysis of parameters. Further, we undertake experiments on UF2, ZDT1, and DTLZ1 with $D = 10$, 20, 30, and 50 to delve into the sensitivity of *NS_int*, enriching our understanding of its effects.

To assess the impact of ensemble membership size, we conducted experiments using various m_{max} values, reaching up to 1000. The

Table 1
The IGD results of SaMOPSO_NS (vs) MOPSO_NS algorithm.

	D = 10			D = 20			D = 30			D = 50		
Funs	SaMOPSO_NS	MOPSO_NS		SaMOPSO_NS	MOPSO_NS		SaMOPSO_NS	MOPSO_NS		SaMOPSO_NS	MOPSO_NS	
UF1	3.6279e-1(9.43e-2)	3.6144e-1(6.04e-2)	–	7.1177e-1(1.23e-1)	6.8792e-1(1.09e-1)	–	9.4536e-1(1.09e-1)	9.2734e-1(1.66e-1)	–	1.1123e + 0(8.22e-2)	1.1809e + 0 (1.05e-1)	+
UF2	1.6988e-1(2.30e-2)	1.7446e-1(1.98e-2)	+	2.7910e-1(3.73e-2)	2.8312e-1(2.35e-2)	+	3.4716e-1(3.03e-2)	3.6189e-1(2.72e-2)	+	4.2140e-1(2.74e-2)	4.0631e-1(3.00e-2)	–
UF3	1.0548e + 0(1.14e-1)	1.1098e + 0 (8.46e-2)	+	8.3611e-1(3.78e-2)	8.4347e-1(4.77e-2)	+	7.2663e-1(6.19e-2)	7.6370e-1(4.58e-2)	+	6.7596e-1(3.56e-2)	7.0656e-1(4.53e-2)	+
UF4	1.4852e-1(8.51e-3)	1.5172e-1(1.09e-2)	+	1.7912e-1(6.74e-3)	1.8355e-1(4.55e-3)	+	1.8452e-1(5.15e-3)	1.9333e-1(4.17e-3)	+	1.9807e-1(4.10e-3)	2.0357e-1(4.03e-3)	+
UF5	2.4254e + 0(3.09e-1)	2.4542e + 0 (3.43e-1)	+	3.0306e + 0(2.24e-1)	3.0257e + 0 (1.37e-1)	–	3.2877e + 0(1.71e-1)	3.3507e + 0 (1.33e-1)	+	3.6714e + 0(1.38e-1)	3.6137e + 0 (1.93e-1)	–
UF6	1.0964e + 0(1.31e-1)	1.2719e + 0 (1.65e-1)	+	1.3084e + 0(2.07e-1)	1.2747e + 0 (1.00e-1)	–	1.3897e + 0(1.95e-1)	1.5092e + 0 (1.98e-1)	+	1.6709e + 0(1.93e-1)	1.6075e + 0 (2.09e-1)	–
UF7	4.7023e-1(9.64e-2)	5.0530e-1(9.43e-2)	+	7.6347e-1(9.85e-2)	7.9565e-1(1.22e-1)	+	8.9927e-1(1.55e-1)	1.0326e + 0 (5.99e-2)	+	1.1983e + 0(8.89e-2)	1.2059e + 0 (1.42e-1)	+
ZDT1	9.4482e-1(1.63e-1)	9.2499e-1(1.40e-1)	=	1.5377e + 0(1.56e-1)	1.5815e + 0 (1.56e-1)	+	1.8212e + 0(1.88e-1)	1.9701e + 0 (1.56e-1)	+	2.1278e + 0(1.31e-1)	2.1180e + 0 (1.89e-1)	–
ZDT2	1.2829e + 0(2.25e-1)	1.4849e + 0 (2.27e-1)	+	2.3194e + 0(1.79e-1)	2.3139e + 0 (1.75e-1)	–	2.7530e + 0(1.81e-1)	2.9148e + 0 (1.40e-1)	+	3.1472e + 0(1.27e-1)	3.1802e + 0 (1.21e-1)	+
ZDT3	8.5904e-1(1.90e-1)	8.1748e-1(1.87e-1)	–	1.4281e + 0(2.03e-1)	1.3122e + 0 (1.44e-1)	–	1.5568e + 0(2.18e-1)	1.6646e + 0 (1.69e-1)	+	1.9454e + 0(2.25e-1)	1.8734e + 0 (1.62e-1)	–
ZDT4	5.1612e + 1(7.61e + 0)	5.5948e + 1(7.27e + 0)	+	1.7632e + 2(1.29e + 1)	1.7573e + 2(1.08e + 1)	–	2.9932e + 2(1.55e + 1)	3.1164e + 2(1.37e + 1)	+	5.5062e + 2(2.64e + 1)	5.6308e + 2(2.17e + 1)	+
ZDT6	5.5751e + 0(4.13e-1)	5.6633e + 0 (3.31e-1)	+	6.4775e + 0(2.28e-1)	6.5949e + 0 (1.57e-1)	+	6.7278e + 0(2.02e-1)	6.8876e + 0 (8.11e-2)	+	7.1781e + 0(7.41e-2)	7.1442e + 0 (1.02e-1)	–
DTLZ1	3.6956e + 1(1.95e + 1)	2.9322e + 1(1.94e + 1)	–	8.2585e + 1(3.15e + 1)	6.1384e + 1(3.49e + 1)	–	1.1968e + 2(5.11e + 1)	1.5429e + 2(7.36e + 1)	+	2.0177e + 2(8.51e + 1)	1.7778e + 2(7.43e + 1)	–
DTLZ2	1.4223e-1(2.68e-2)	1.3704e-1(1.45e-2)	–	4.5144e-1(6.66e-2)	4.6298e-1(6.46e-2)	+	8.1386e-1(5.87e-2)	8.0646e-1(7.06e-2)	–	1.4475e + 0(1.13e-1)	1.5417e + 0 (9.29e-2)	+
DTLZ3	5.3537e + 2(9.17e + 1)	5.4044e + 2(4.58e + 1)	=	3.4835e + 2(1.12e + 2)	3.6712e + 2(1.00e + 2)	+	2.3774e + 2(7.93e + 1)	1.7171e + 2(1.28e + 2)	–	1.9110e + 2(9.19e + 1)	1.1880e + 2(8.55e + 1)	–
DTLZ4	7.1299e-1(1.06e-1)	6.8684e-1(1.07e-1)	–	8.7729e-1(1.17e-1)	9.1579e-1(8.46e-2)	+	1.1726e + 0(6.13e-2)	1.2359e + 0 (7.37e-2)	+	1.8021e + 0(2.10e-1)	2.0401e + 0 (1.03e-1)	+
+/=/-	**9/2/5**			**9/07**			**13/0/3**			**8/0/8**		

Table 2
The IGD results of SaMOPSO_NS (vs) SaMOPSO algorithm.

	D = 10			D = 20			D = 30			D = 50		
Funs	SaMOPSO_NS	SaMOPSO		SaMOPSO_NS	SaMOPSO		SaMOPSO_NS	SaMOPSO		SaMOPSO_NS	SaMOPSO	
UF1	3.8907e-1(9.96e-2)	3.4866e-1(7.00e-2)	–	6.5971e-1(8.74e-2)	7.2671e-1(1.59e-1)	+	9.1225e-1(1.13e-1)	8.7218e-1(8.39e-2)	–	1.0928e + 0(8.52e-2)	1.1222e + 0 (1.39e-1)	+
UF2	1.6833e-1(2.44e-2)	1.7066e-1(2.10e-2)	+	2.9173e-1(3.01e-2)	2.9390e-1(3.01e-2)	+	3.3831e-1(4.01e-2)	3.4336e-1(2.75e-2)	+	4.1485e-1(2.12e-2)	4.3987e-1(3.41e-2)	+
UF3	9.9739e-1(1.38e-1)	1.0643e + 0 (9.91e-2)	+	8.2862e-1(4.94e-2)	8.5499e-1(4.43e-2)	+	7.2123e-1(4.42e-2)	7.6350e-1(6.63e-2)	+	6.6089e-1(3.68e-2)	6.9947e-1(1.98e-2)	+
UF4	1.4858e-1(5.83e-3)	1.6003e-1(3.74e-3)	+	1.7721e-1(8.27e-3)	1.8262e-1(5.86e-3)	+	1.8889e-1(2.85e-3)	1.9359e-1(4.60e-3)	+	1.9817e-1(2.58e-3)	2.0582e-1(2.84e-3)	+
UF5	2.3811e + 0(3.64e-1)	2.5337e + 0 (3.70e-1)	+	3.1190e + 0(2.36e-1)	3.1244e + 0 (2.49e-1)	–	3.3660e + 0(2.18e-1)	3.3067e + 0 (1.89e-1)	–	3.5982e + 0(1.94e-1)	3.6571e + 0 (1.54e-1)	+
UF6	1.2479e + 0(1.67e-1)	1.2685e + 0 (2.11e-1)	+	1.3027e + 0(2.25e-1)	1.2554e + 0 (1.94e-1)	–	1.3678e + 0(1.54e-1)	1.5143e + 0 (2.42e-1)	+	1.6753e + 0(2.50e-1)	1.7020e + 0 (2.40e-1)	–
UF7	4.3744e-1(1.10e-1)	4.6587e-1(6.72e-2)	+	7.5426e-1(1.23e-1)	7.5889e-1(1.51e-1)	+	9.9280e-1(1.37e-1)	9.8648e-1(1.46e-1)	=	1.2001e + 0(9.06e-2)	1.2354e + 0 (9.85e-2)	–
ZDT1	1.0199e + 0(1.66e-1)	9.4017e-1(1.60e-1)	–	1.5447e + 0(1.96e-1)	1.6309e + 0 (1.44e-1)	+	1.7571e + 0(1.34e-1)	1.8770e + 0 (1.39e-1)	+	2.1095e + 0(9.11e-2)	2.1946e + 0 (1.42e-1)	+
ZDT2	1.3803e + 0(1.73e-1)	1.4375e + 0 (1.54e-1)	+	2.1202e + 0(3.79e-1)	2.1590e + 0 (2.74e-1)	+	2.6760e + 0(2.07e-1)	2.6495e + 0 (3.15e-1)	–	3.1310e + 0(1.71e-1)	3.2003e + 0 (1.64e-1)	+
ZDT3	8.6600e-1(1.75e-1)	9.3960e-1(1.52e-1)	+	1.3907e + 0(2.37e-1)	1.4206e + 0 (1.86e-1)	+	1.6056e + 0(1.92e-1)	1.6207e + 0 (1.12e-1)	+	1.8066e + 0(1.08e-1)	1.9046e + 0 (1.20e-1)	+
ZDT4	6.1228e + 1(5.90e + 0)	5.8524e + 1(4.29e + 0)	–	1.6702e + 2(9.46e + 0)	1.7607e + 2(1.22e + 1)	+	2.9932e + 2(1.55e + 1)	3.0844e + 2(1.17e + 1)	+	5.5062e + 2(2.64e + 1)	5.6823e + 2(2.45e + 1)	+
ZDT6	5.5949e + 0(3.63e-1)	5.6307e + 0 (2.82e-1)	+	6.5739e + 0(1.89e-1)	6.5344e + 0 (1.50e-1)	–	6.8749e + 0(1.53e-1)	6.8706e + 0 (1.26e-1)	–	7.1266e + 0(1.40e-1)	7.1335e + 0 (1.03e-1)	+
DTLZ1	2.9382e + 1(2.13e + 1)	3.0665e + 1(2.21e + 1)	=	7.2236e + 1(6.24e + 1)	6.2294e + 1(4.48e + 1)	–	1.0874e + 2(4.44e + 1)	1.3776e + 2(7.18e + 1)	+	1.8201e + 2(9.82e + 1)	1.7724e + 2(7.73e + 1)	–
DTLZ2	1.4394e-1(3.15e-2)	1.6038e-1(1.50e-2)	+	4.5474e-1(6.79e-2)	4.6763e-1(5.27e-2)	+	7.5984e-1(5.61e-2)	8.2630e-1(9.35e-2)	+	1.5147e + 0(1.43e-1)	1.5643e + 0 (6.56e-2)	+
DTLZ3	5.3574e + 2(5.36e + 1)	5.6479e + 2(6.02e + 1)	+	3.3510e + 2(9.10e + 1)	3.6649e + 2(1.02e + 2)	+	2.2972e + 2(1.15e + 2)	2.5070e + 2(8.87e + 1)	+	1.2655e + 2(7.00e + 1)	1.4448e + 2(9.53e + 1)	+
DTLZ4	7.3236e-1(9.42e-2)	6.6259e-1(1.66e-1)	–	9.0417e-1(3.36e-2)	9.3453e-1(5.70e-2)	+	1.1118e + 0(7.11e-2)	1.1751e + 0 (9.75e-2)	+	1.8448e + 0(2.04e-1)	1.8901e + 0 (1.86e-1)	+
+/=/-	**11/1/4**			**12/0/4**			**11/1/4**			**13/0/3**		

average error over 21 independent runs is presented in Fig. 3, with the red, blue, and black lines corresponding to decision variables of sizes 10, 30, and 50, respectively. Due to space constraints, the legend is regrettably omitted in Fig. 3. Interestingly, we observed that the error profiles initially decrease with increasing ensemble size for both unimodal and multimodal problems. However, this decline becomes sluggish or even slightly reversed when the size exceeds 10. Furthermore, our experiments revealed a linear growth in computational costs as the ensemble size increases, likely driven by increased modeling time. In other words, the larger the model size, the more time-consuming it is. These findings align with those reported in [34], although the distinction lies in the ensemble size being set to 100 in that study. Moreover, a significant insight from our study is the superior performance of ensemble surrogates (model size >1) over a singular surrogate (model size $= 1$), which justifies the use of ensemble surrogates in this article. Given these findings, we determined an ensemble membership size of $m_{max} = 10$ for this paper, balancing the trade-off between approximation accuracy and computational efficiency effectively.

Next, to investigate the effect of the generation interval on our proposed algorithm, we varied *NS_int* values, specifically *NS_int* $\in\{1, 2, 3, 4, 5\}$, and tested them on three representative instances for each category. If no predicted non-dominated individuals satisfy the *pbest*-dominated infill criterion, all available FEs are exhausted for neighborhood search. Based on the configured parameters, the maximum generation interval can be calculated as *maxFES/n_EXA* $= 6$. Consequently, the proposed infill criterion will not be executed when *NS_int* >5 due to the lack of available FEs. Therefore, we set the upper limit of parameter *NS_int* to 5 in this experiment. Fig. 4 presents the IGD results of SaMOPSO_NS with different *NS_int* values, obtained from 20 independent runs. Observations from this figure indicate that the best results on three issues were achieved when *NS_int* $= 2$. Simultaneously, the second-best results were secured when *NS_int* $= 4$. As per our previous explanation, an overly large or small *NS_int* value does little to benefit the SaMOPSO_NS algorithm due to the unbalanced allocation of computational resources. As expected, we find that the IGD is somewhat inferior when *NS_int* $= 1$ and *NS_int* $= 5$ in Fig. 4. Moreover, we notice that the boxplot of IGD has a large area when *NS_int* $= 3$, particularly for UF2 and DTLZ1. This indicates that the algorithm lacks robustness. Considering the aforementioned evidence, we recommend *NS_int* be set at 2 or 4, with a standardized setting of *NS_int* $= 2$ for all test instances in this study.

4.3. Behavioral effects of SaMOPSO_NS

1) *Effects of the pbest-Dominated Infill Criterion*: In this subsection, we aim to investigate the effects of the *pbest*-dominated infill criterion by comparing MOPSO_NS, a newly modified variant algorithm that does not employ a surrogate, with SaMOPSO_NS. Unlike SaMOPSO_NS, MOPSO_NS evaluates all offspring individuals, not just the predicted non-dominated ones. We conducted a comparison of the two algorithms on UF, ZDT, and DTLZ test instances with 21 independent runs. Table 1 provides the mean values and standard deviations of the obtained IGD values. The Wilcoxon rank-sum test was applied to conduct a significance test in comparing SaMOPSO_NS with MOPSO_NS at $\alpha = 0.95$ significance level, where symbols '+', '=', and '−' indicate that the SaMOPSO_NS is statistically better than, similar to, and statistically worse than its competitor, respectively.

From Table 1, it can be observed that SaMOPSO_NS outperforms MOPSO_NS significantly when $D = 10$ and 30, slightly outperforms when $D = 20$, and shows similar performance when $D = 50$. Specifically, when $D = 10$, the rank-sum test result of SaMOPSO_NS vs MOPSO_NS algorithm is 9/2/5, indicating that SaMOPSO_NS performs better than MOPSO_NS on 9 test problems, shows similar performance on 2 test problems, and performs statistically worse on 5 test problems. When $D = 20$, 30, and 50, SaMOPSO_NS and MOPSO_NS algorithms did not show similar performance on any of the test problems. Instead, SaMOPSO_NS outperforms MOPSO_NS on 9, 13, and 8 test problems for $D = 20$, 30, and 50, respectively. It is worth mentioning that SaMOPSO_NS and MOPSO_NS perform equally when D = 50, as they show better performance than each other on 8 test problems. Overall, SaMOPSO_NS does not convey significant disadvantages compared to the MOPSO_NS algorithm on different dimensions. The remarkable success of the SaMOPSO_NS algorithm can be attributed to two factors. Firstly, the proposed infill criterion effectively facilitates the algorithm in utilizing limited computing resources to their maximum potential. Secondly, the improved EXA would enable the global best individuals to guide the population to search for some not-well-explored areas. Experimental findings suggest that SaMOPSO_NS emerges as the victor in this competition, displaying a marked superiority over MOPSO_NS in the majority of the test cases. Based on this evidence, it can be inferred that the novel infill criterion introduced in this paper is efficacious in enhancing algorithmic performance.

2) *Effects of the EXA-NS Strategy*: This subsection examines the effect of the EXA-NS strategy on the SaMOPSO_NS algorithm's performance. 21 independent runs were conducted for both the SaMOPSO_NS and SaMOPSO algorithms, using UF, ZDT, and DTLZ test problems with dimensions D = 10, 20, 30, and 50. Within the proposed algorithm, the EXA-NS strategy comes into effect only when the IGD value remains consistent across successive *NS_int* generations. Upon meeting this condition, it suggests that the current population might be stagnant and even trapped in a local optimum. The EXA-NS strategy aims to expedite population evolution towards the true PF and enhance global search capability. The experimental results, including mean values, standard deviations, and the rank-sum test of the IGD values acquired by the two algorithms, are presented in Table 2.

The statistical result of SaMOPSO_NS vs SaMOPSO observed from Table 2 is 11/1/4 for $D = 10$ and $D = 30$, which implies that SaMOPSO_NS performs better than, similar to, and statistically worse than SaMOPSO algorithm on 11, 1, and 4 test problems, respectively. When $D = 20$ and $D = 50$, the statistical result is 12/0/4 and 13/0/3, respectively, indicating that SaMOPSO_NS is significantly outstanding than SaMOPSO algorithm on 12 and 13 test problems respectively, and only significantly inferior to SaMOPSO on 4 and 3 test problems. This analysis demonstrates that the incorporated EXA-NS strategy improves the original algorithm's competency in handling medium-dimensional MOPs, albeit not universally. The enhanced performance of the SaMOPSO_NS in

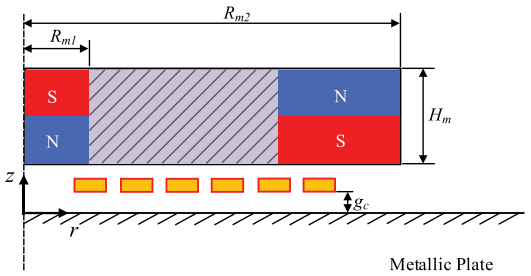


Fig. 5. Convergence profiles of IDG values of SaMOPSO_NS (vs) SaMOPSO. Note: 'A1′ and 'A2′ represent SaMOPSO_NS and SaMOPSO respectively.

Table 3
The IGD results of the six algorithms on all test instances with $D = 10$.

Funs(D = 10)	SaMOPSO_NS	CSEA		ParEGO		MOEA/D-EGO		KRVEA		EIMEGO	
UF1	3.3481e-1(8.67e-2)	2.5515e-1 (8.79e-2)	=	2.1956e-1 (3.52e-2)	–	2.5407e-1 (9.39e-2)	–	1.3085e-1 (4.02e-2)	–	**1.2559e-1 (3.03e-2)**	–
UF2	1.5807e-1(2.24e-2)	1.6052e-1 (4.21e-2)	=	2.6647e-1 (8.38e-2)	+	2.2108e-1 (3.52e-2)	+	7.2817e-2 (1.04e-2)	–	**7.2321e-2 (8.93e-3)**	–
UF3	1.0555e + 0 (1.25e-1)	1.1793e + 0 (1.69e-1)	=	1.1370e + 0 (1.79e-1)	=	9.3749e-1 (1.76e-1)	=	**8.0749e-1 (1.86e-1)**	–	1.0228e + 0 (1.14e-1)	=
UF4	1.5913e-1(5.88e-3)	1.3066e-1 (5.88e-3)	–	9.0738e-2 (1.10e-2)	–	1.1015e-1 (1.14e-2)	–	1.0297e-1 (1.15e-2)	–	**7.3822e-2 (5.08e-3)**	–
UF5	2.4121e + 0 (2.55e-1)	2.1929e + 0 (4.56e-1)	=	2.2727e + 0 (5.05e-1)	=	3.0488e + 0 (4.96e-1)	+	**1.1482e + 0 (2.96e-1)**	–	2.6813e + 0 (6.29e-1)	=
UF6	1.1673e + 0 (2.44e-1)	2.0513e + 0 (1.93e-1)	+	1.4908e + 0 (3.55e-1)	+	1.7262e + 0 (5.16e-1)	+	**1.1373e + 0 (1.98e-1)**	=	1.5559e + 0 (1.45e-1)	+
UF7	4.7738e-1(7.17e-2)	4.0400e-1 (1.14e-1)	=	3.1050e-1 (6.55e-2)	–	2.9930e-1 (9.16e-2)	–	2.2064e-1 (6.23e-2)	–	**2.1853e-1 (5.42e-2)**	–
ZDT1	1.0075e + 0 (1.95e-1)	2.5335e-1 (1.83e-1)	–	**2.4702e-2 (6.70e-3)**	–	7.8330e-2 (4.93e-2)	–	3.8182e-2 (1.03e-2)	–	1.3043e-1 (4.06e-2)	–
ZDT2	1.3177e + 0 (9.59e-2)	7.9238e-1 (1.80e-1)	–	**2.3103e-2 (4.33e-3)**	–	6.8514e-2 (7.34e-2)	–	3.4367e-2 (3.85e-3)	–	4.1291e-1 (2.36e-1)	–
ZDT3	8.7665e-1(1.60e-1)	1.8616e-1 (6.13e-2)	–	5.8359e-2 (1.09e-2)	–	3.2275e-1 (1.35e-1)	–	**5.5114e-2 (9.38e-3)**	–	9.2669e-2 (1.98e-2)	–
ZDT4	5.4944e + 1 (8.81e + 0)	4.5458e + 1 (1.00e + 1)	–	6.0144e + 1 (1.46e + 1)	=	8.1427e + 1 (1.38e + 1)	+	**3.4705e + 1 (1.49e + 1)**	–	5.8358e + 1 (1.03e + 1)	=
ZDT6	5.6206e + 0 (3.90e-1)	4.3917e + 0 (6.64e-1)	–	**3.6430e-1 (1.11e-1)**	–	1.2777e + 0 (7.67e-1)	–	9.7275e-1 (1.39e-1)	–	6.0681e-1 (3.92e-1)	–
DTLZ1	**3.1519e + 1 (2.02e + 1)**	8.3641e + 1 (2.19e + 1)	+	8.1880e + 1 (8.81e + 0)	+	1.1395e + 2 (2.23e + 1)	+	1.0333e + 2 (2.61e + 1)	+	1.3024e + 2 (1.71e + 1)	+
DTLZ2	1.3768e-1(1.41e-2)	1.1862e-1 (3.25e-2)	–	1.4509e-1 (2.26e-2)	=	2.6961e-1 (4.25e-2)	+	**3.4978e-2 (4.49e-3)**	–	8.8810e-2 (1.06e-2)	–
DTLZ3	5.6761e + 2 (4.50e + 1)	2.2098e + 2 (5.79e + 1)	–	**1.9764e + 2 (6.13e + 0)**	–	2.7091e + 2 (7.01e + 1)	–	2.4888e + 2 (9.43e + 1)	–	2.8799e + 2 (6.15e + 1)	–
DTLZ4	6.6585e-1(1.58e-1)	3.5245e-1 (1.36e-1)	–	2.7299e-1 (1.13e-1)	–	5.1395e-1 (6.42e-2)	–	**1.8305e-1 (1.97e-1)**	–	4.6382e-1 (2.15e-1)	–
+/=/-	**NA**	**2/5/9**		**3/4/9**		**6/1/9**		**1/1/14**		**2/3/11**	

comparison to SaMOPSO, which lacks the EXA-NS strategy, might be attributed to the former's greater emphasis on refining local exploitation around individuals captured in the EXA. It is each individual in the updated EXA, as a global best particle, that leads the other particles to evolve towards a more promising PS step by step. In summary, these observations collectively substantiate that the EXA-NS strategy significantly boosts the algorithm's solution accuracy.

In Fig. 5, convergence profiles of the mean IGD values obtained by the two algorithms are presented for all functions, except for DTLZ1, due to its multimodal nature and the presence of numerous influential points. The UF suite, derived from the CEC'09 benchmark problems, is composed of a collection of general unconstrained (bound-constrained) two-objective test functions. Within this suite, ZDT1 and ZDT2 serve as fundamental MOPs to evaluate the algorithm's ability to exploit the PF's shape. On the other hand, ZDT3, ZDT4, and ZDT6 are complicated MOPs. ZDT3 exhibits a non-continuous PF, while ZDT4 and ZDT6 represent MOPs with multiple models. In the DTLZ suite, all decision variables fall within the range [0,1]. Notably, challenges arise for SAEAs to converge to the global PF due to the presence of numerous local PFs in DTLZ1 and DTLZ3.

As shown in Fig. 5, one evident observation is that SaMOPSO_NS converges faster than SaMOPSO on most functions except a few. Especially for UF4 and ZDT3, SaMOPSO_NS's convergence speed significantly outpaces that of SaMOPSO. This finding suggests the EXA-NS strategy's effectiveness in accelerating local exploitation of the population and enhancing algorithm convergence. Moreover,

Table 4
The IGD results of the six algorithms on all test instances with $D = 20$.

Funs($D = 20$)	SaMOPSO_NS	CSEA		ParEGO		MOEA/D-EGO		KRVEA		EIMEGO	
UF1	7.2002e-1(7.45e-2)	6.8793e-1 (1.62e-1)	=	8.3156e-1 (1.95e-1)	=	8.2335e-1 (1.91e-1)	=	**3.4195e-1 (1.10e-1)**	–	7.4932e-1 (1.57e-1)	=
UF2	3.0132e-1(2.96e-2)	4.2298e-1 (5.84e-2)	+	4.7015e-1 (7.59e-2)	+	3.6573e-1 (4.18e-2)	+	**1.6404e-1 (3.13e-2)**	–	1.7006e-1 (1.47e-2)	–
UF3	8.6598e-1(4.67e-2)	9.7031e-1 (8.64e-2)	+	1.0900e + 0 (9.97e-2)	+	8.6902e-1 (8.24e-2)	=	**6.7587e-1 (2.79e-2)**	–	7.1655e-1 (4.20e-2)	–
UF4	1.8403e-1(3.48e-3)	1.6684e-1 (4.29e-3)	–	1.3873e-1 (1.50e-2)	–	1.4911e-1 (6.75e-3)	–	1.6452e-1 (5.84e-3)	–	**1.2264e-1 (4.74e-3)**	–
UF5	3.1782e + 0 (2.21e-1)	4.1865e + 0 (3.32e-1)	+	4.3052e + 0 (4.93e-1)	+	4.3086e + 0 (3.92e-1)	+	**3.1302e + 0 (4.23e-1)**	=	3.9447e + 0 (2.33e-1)	+
UF6	**1.3347e + 0 (2.75e-1)**	3.5791e + 0 (6.69e-1)	+	3.8964e + 0 (9.09e-1)	+	3.6087e + 0 (1.14e + 0)	+	1.6327e + 0 (6.38e-1)	=	2.9243e + 0 (2.64e-1)	+
UF7	7.9269e-1(1.44e-1)	7.3933e-1 (1.90e-1)	=	9.0737e-1 (2.44e-1)	=	8.4199e-1 (1.49e-1)	=	**5.1203e-1 (2.20e-1)**	–	7.3799e-1 (1.51e-1)	=
ZDT1	1.5166e + 0 (1.51e-1)	1.0802e + 0 (2.61e-1)	–	**1.3419e-1 (8.06e-2)**	–	1.4383e + 0 (2.96e-1)	=	1.5306e-1 (5.91e-2)	–	6.5093e-1 (1.31e-1)	–
ZDT2	2.1275e + 0 (1.81e-1)	2.2413e + 0 (2.75e-1)	=	**7.8778e-2 (1.16e-2)**	–	2.4888e + 0 (5.53e-1)	+	2.3054e-1 (1.99e-1)	–	1.4120e + 0 (5.94e-1)	–
ZDT3	1.4080e + 0 (2.03e-1)	8.9591e-1 (2.08e-1)	–	**1.5717e-1 (4.42e-2)**	–	1.4028e + 0 (1.34e-1)	=	6.3045e-1 (4.64e-1)	–	8.9150e-1 (2.31e-1)	–
ZDT4	1.7836e + 2(9.68e + 0)	1.8482e + 2 (2.32e + 1)	=	1.9183e + 2 (1.27e + 1)	+	2.1985e + 2 (1.33e + 1)	+	1.5868e + 2 (3.18e + 1)	–	**1.5797e + 2 (2.26e + 1)**	–
ZDT6	6.4499e + 0 (1.68e-1)	6.6736e + 0 (1.71e-1)	+	**2.1252e + 0 (3.57e-1)**	–	6.8666e + 0 (1.96e-1)	+	4.4302e + 0 (1.65e + 0)	–	3.7094e + 0 (5.90e-1)	–
DTLZ1	**7.5202e + 1 (4.87e + 1)**	3.6250e + 2 (5.70e + 1)	+	2.3858e + 2 (1.06e + 1)	+	4.1676e + 2 (6.67e + 1)	+	4.1510e + 2 (3.72e + 1)	+	3.2611e + 2 (2.90e + 1)	+
DTLZ2	4.6587e-1(4.14e-2)	6.6332e-1 (5.68e-2)	+	6.0645e-1 (1.39e-1)	+	6.5046e-1 (5.57e-2)	+	4.8373e-1 (1.28e-1)	=	**4.3277e-1 (3.38e-2)**	–
DTLZ3	**3.3475e + 2 (9.99e + 1)**	1.0059e + 3 (1.14e + 2)	+	5.0633e + 2 (3.34e + 1)	+	8.8263e + 2 (2.31e + 2)	+	1.0736e + 3 (1.32e + 2)	+	7.5665e + 2 (9.30e + 1)	+
DTLZ4	8.8609e-1(9.64e-2)	8.6221e-1 (1.45e-1)	=	7.9047e-1 (3.16e-1)	=	1.0605e + 0 (1.01e-1)	+	9.3619e-1 (1.75e-1)	=	**7.5935e-1 (7.10e-2)**	–
+/=/-	**NA**	**8/7/1**		**8/3/5**		**10/5/1**		**2/4/10**		**4/2/10**	

it can be observed that the convergence processes of the two algorithms display remarkable similarity in certain test instances such as UF1($D = 10, 30$), UF2($D = 10, 30$), UF6($D = 20, 50$), UF7($D = 20, 30$), ZDT6($D = 20, 50$), and DTLZ3($D = 50$). Although SaMOPSO_NS converges slower only on UF5 with $D = 30$, it manages to yield satisfactory results in the early stages. The slower convergence of SaMOPSO_NS compared to SaMOPSO may be attributed to SaMOPSO_NS likely being caught in local optima during the latter stage. On the other hand, an evident disparity reflected in the figure is that SaMOPSO has evolved over more generations compared to SaMOPSO_NS. A plausible explanation is that with the equal allocation of FEs, SaMOPSO_NS consumes more FEs per generation compared to SaMOPSO_NS, leading to a halt in its evolution after fewer generations. All these observations serve to demonstrate the efficacy of the EXA-NS strategy.

In summary, the aforementioned experimental findings and analysis corroborate the enhanced performance of algorithms employing both the *pbest*-dominated infill criterion and the EXA-NS strategy, compared to utilizing one of them in isolation. Both of them play a positive and complementary role in the SaMOPSO_NS algorithm. The proposed criterion, as a pre-screening strategy, can effectively reduce some inessential expensive FEs on unreliable individuals, thereby reserving more computational resources for local searches. The newly proposed EXA-NS strategy concentrates on the current local fitness landscape, directing the swarm towards promising areas. The definitive superiority of SaMOPSO_NS is largely attributed to the collaborative efforts of the proposed criterion and EXA-NS strategy, which strategically manage and optimize the usage of limited computational resources.

4.4. Compare with other SAEAs

To investigate the performance of the SaMOPSO_NS algorithm in solving expensive MOPs with different dimensions, we compare it with several popular surrogate-assisted MOEAs on all test instances in this subsection. We, once again, employ the Wilcoxon rank-sum test to execute a statistical significance comparison between SaMOPSO_NS and other algorithms. Table 3 to Table 4 present the mean and standard deviation of IGD values for the six algorithms based on 21 independent runs for dimensions $D = 10$, 20, 30, and 50. The best values are prominently highlighted. Throughout this paper, we classify MOPs with dimensions $D = 10$, 20, as low-dimensional and those with $D = 30$, 50 as medium-dimensional.

From all tables, it is noticeable that SaMOPSO_NS exhibits a significant performance differentiation on varying dimensional instances. Specifically,

When $D = 10$ and $D = 20$, the number of functions where SaMOPSO_NS outperforms its five rivals equals 2/3/6/1/2 and 8/8/10/2/

Table 5
The IGD results of the six algorithms on all test instances with $D = 30$.

Funs(D = 30)	SaMOPSO_NS	CSEA		ParEGO		MOEA/D-EGO		KRVEA		EIMEGO	
UF1	**9.2015e-1(1.53e-1)**	1.2478e + 0 (1.23e-1)	+	1.2017e + 0 (1.35e-1)	+	1.2806e + 0 (9.10e-2)	+	1.2323e + 0 (1.14e-1)	+	1.2608e + 0 (1.22e-1)	+
UF2	**3.6923e-1(2.34e-2)**	5.8291e-1 (3.05e-2)	+	5.8838e-1 (1.73e-2)	+	5.8518e-1 (3.83e-2)	+	5.8846e-1 (5.74e-2)	+	5.9452e-1 (2.50e-2)	+
UF3	**7.3890e-1(3.07e-2)**	1.1032e + 0 (4.66e-2)	+	1.0538e + 0 (7.94e-2)	+	1.0705e + 0 (7.31e-2)	+	1.0511e + 0 (6.39e-2)	+	1.1137e + 0 (8.50e-2)	+
UF4	1.9429e-1(4.88e-3)	1.8650e-1 (3.80e-3)	–	1.8631e-1 (3.65e-3)	–	**1.8372e-1 (3.53e-3)**	–	1.8636e-1 (4.28e-3)	–	1.8624e-1 (3.59e-3)	–
UF5	**3.4891e + 0 (2.59e-1)**	5.0845e + 0 (3.74e-1)	+	4.9421e + 0 (3.42e-1)	+	5.0698e + 0 (2.14e-1)	+	4.9472e + 0 (3.04e-1)	+	5.2359e + 0 (2.81e-1)	+
UF6	**1.5103e + 0 (2.53e-1)**	5.1937e + 0 (5.30e-1)	+	4.9373e + 0 (5.22e-1)	+	5.2841e + 0 (3.62e-1)	+	5.2265e + 0 (4.53e-1)	+	5.1509e + 0 (5.84e-1)	+
UF7	**9.0596e-1(1.15e-1)**	1.2513e + 0 (8.83e-2)	+	1.2625e + 0 (1.45e-1)	+	1.3109e + 0 (1.79e-1)	+	1.2578e + 0 (7.69e-2)	+	1.3097e + 0 (1.03e-1)	+
ZDT1	**1.9354e + 0 (1.14e-1)**	2.0966e + 0 (7.46e-2)	+	2.0515e + 0 (1.87e-1)	=	2.1168e + 0 (1.26e-1)	+	2.0571e + 0 (1.24e-1)	+	2.1222e + 0 (1.24e-1)	+
ZDT2	**2.7492e + 0 (1.96e-1)**	3.4168e + 0 (1.03e-1)	+	3.3852e + 0 (1.98e-1)	+	3.4025e + 0 (1.47e-1)	+	3.3880e + 0 (1.57e-1)	+	3.3540e + 0 (1.71e-1)	+
ZDT3	**1.6184e + 0 (1.80e-1)**	1.8057e + 0 (1.93e-1)	+	1.7884e + 0 (1.55e-1)	+	1.7253e + 0 (1.45e-1)	=	1.7444e + 0 (1.60e-1)	=	1.7368e + 0 (1.18e-1)	=
ZDT4	**3.0048e + 2 (2.15e + 1)**	3.6489e + 2 (1.87e + 1)	+	3.6436e + 2 (1.72e + 1)	+	3.5431e + 2 (1.62e + 1)	+	3.6398e + 2 (2.05e + 1)	+	3.6888e + 2 (1.66e + 1)	+
ZDT6	**6.9005e + 0 (1.53e-1)**	7.2170e + 0 (8.42e-2)	+	7.2764e + 0 (9.29e-2)	+	7.2408e + 0 (7.63e-2)	+	7.2258e + 0 (8.90e-2)	+	7.2349e + 0 (7.48e-2)	+
DTLZ1	**1.3650e + 2 (1.23e + 2)**	7.7542e + 2 (5.63e + 1)	+	7.7149e + 2 (4.14e + 1)	+	7.8610e + 2 (5.51e + 1)	+	7.7995e + 2 (4.29e + 1)	+	7.6954e + 2 (5.57e + 1)	+
DTLZ2	**8.0792e-1(6.96e-2)**	1.4621e + 0 (9.15e-2)	+	1.4632e + 0 (7.41e-2)	+	1.4639e + 0 (6.48e-2)	+	1.4290e + 0 (9.75e-2)	+	1.4656e + 0 (1.27e-1)	+
DTLZ3	**2.6463e + 2 (7.75e + 1)**	2.0105e + 3 (1.01e + 2)	+	2.0533e + 3 (1.80e + 2)	+	2.0758e + 3 (1.06e + 2)	+	2.0736e + 3 (1.54e + 2)	+	2.0471e + 3 (7.98e + 1)	+
DTLZ4	**1.1933e + 0 (1.00e-1)**	1.7659e + 0 (1.03e-1)	+	1.7664e + 0 (8.43e-2)	+	1.7935e + 0 (7.88e-2)	+	1.7170e + 0 (1.08e-1)	+	1.7645e + 0 (1.25e-1)	+
+/=/-	**NA**	**15/0/1**		**14/1/1**		**14/1/1**		**14/1/1**		**14/1/1**	

4, respectively. The number rises to 15/14/14/14/14 and 14/14/13/13/13 when $D = 30$ and $D = 50$, respectively. It is abundantly clear that SaMOPSO_NS achieves markedly superior results as the dimension increases, thus suggesting that the SaMOPSO_NS algorithm surpasses other algorithms in handling medium-dimensional problems. Our primary focus is to tackle expensive MOPs in medium dimensions in this paper. It is worth noting that the reported surrogate-assisted MOEAs having been able to solve the expensive MOPs with medium dimension well, but the dimension is only up to 30.

When $D = 10$, CSEA and MOEA/D-EGO do not obtain the overall best IGD value on any function. SaMOPSO_NS obtains the overall best IGD value only on the DTLZ1 function. Both ParEGO and EIMEGO obtain four overall best IGD values on ZDT1, ZDT2, ZDT6, DTLZ3 and UF1, UF2, UF4, UF7 respectively. KRVEA obtained the largest number of overall best IGD values on seven functions. A similar observation can be found when $D = 20$ as well. The only difference lies in the fact that, for $D = 20$, the number of overall best IGD values obtained by SaMOPSO_NS has increased while that of EIMEGO has decreased. When it comes to MPOs with $D = 10$, the other five algorithms exhibit overwhelming advantages over SaMOPSO_NS. This may be attributed to the factor that MOPs with $D = 10$ are relatively simpler compared to those with higher dimensions, and existing surrogate-assisted MOEAs have proven to be highly proficient in solving these problems. Compared to $D = 10$, we can find that the superiority of SaMOPSO_NS becomes somewhat prominent when $D = 20$. As shown in Table 4, SaMOPSO_NS outperforms CSEA, ParEGO, and MOEA/D-EGO yet falls short of KRVEA and EIMEGO. KRVEA and EIMEGO each win ten out of sixteen functions and lose just two and four, respectively. In each generation, KRVEA employed two distinct criteria to select individual for real evaluation. EIMEGO employs a Kriging model to approximate each objective function, and the candidate individuals with the highest value under the maximin distance-based EIM criterion are exactly evaluated. However, the EIM criterion is not suitable for high-dimensional optimization problems, given that the search space experiences an exponential growth with increasing dimensions, as evidenced by Table 5 and Table 6.

As shown in Table 5, SaMOPSO_NS significantly outperformed ParEGO, MOEADEGO, KRVEA, and EIMEGO on 14 test instance in each comparison, which may indicate the superiority of the proposed criterion. SaMOPSO NS is statistically inferior to other algorithms only on UF4, comparable to ParEGO on ZDT1, and similar to MOEA/D-EGO, KRVEA, and EIMEGO on ZDT3, and better than all compared algorithms on the remaining functions. According to Table 6, compared to CSEA and ParEGO, SaMOPSO_NS obtains the same statistical results, that is 14+, 1=, 1-. Compared to MOEA/D-EGO, KRVEA, and EIMEGO, SaMOPSO_NS also obtain the same IGD statistical results, that is, 13+, 2=, 1-. The overwhelming superiority of SaMOPSO_NS presents that it beats all the compared SAEAs on 14 and 13 functions. SaMOPSO_NS is tied with all SAEAs on ZDT3, and tied with MOEA/D-EGO, KRVEA, EIMEGO on ZDT1 respectively, while only loses the competition only on UF4. In addition, it is found that SaMOPSO_NS is inferior to all its opponents on UF4 in

Table 6
The IGD results of the six algorithms on all test instances with $D = 50$.

Funs(D = 50)	SaMOPSO_NS	CSEA		ParEGO		MOEA/D-EGO		KRVEA		EIMEGO	
UF1	**1.2114e + 0 (8.08e-2)**	1.4401e + 0 (8.46e-2)	+	1.4104e + 0 (1.20e-1)	+	1.4217e + 0 (7.41e-2)	+	1.4174e + 0 (9.61e-2)	+	1.4196e + 0 (8.98e-2)	+
UF2	**4.3921e-1(3.79e-2)**	6.7308e-1 (2.95e-2)	+	6.6692e-1 (3.06e-2)	+	6.6594e-1 (3.31e-2)	+	6.6174e-1 (2.20e-2)	+	6.5723e-1 (1.98e-2)	+
UF3	**6.9069e-1(3.55e-2)**	1.0394e + 0 (3.02e-2)	+	1.0356e + 0 (4.09e-2)	+	9.8965e-1 (4.92e-2)	+	1.0070e + 0 (4.07e-2)	+	1.0052e + 0 (2.93e-2)	+
UF4	2.0614e-1(2.42e-3)	1.9225e-1 (2.26e-3)	–	1.9321e-1 (2.66e-3)	–	**1.9149e-1 (2.19e-3)**	–	1.9175e-1 (2.65e-3)	–	1.9367e-1 (1.89e-3)	–
UF5	**3.5856e + 0 (2.40e-1)**	5.4239e + 0 (2.22e-1)	+	5.4898e + 0 (1.64e-1)	+	5.4902e + 0 (2.28e-1)	+	5.6331e + 0 (2.39e-1)	+	5.5216e + 0 (2.67e-1)	+
UF6	**1.7329e + 0 (1.53e-1)**	5.8441e + 0 (3.29e-1)	+	6.0144e + 0 (5.03e-1)	+	5.9782e + 0 (3.97e-1)	+	6.0171e + 0 (5.50e-1)	+	5.7730e + 0 (3.95e-1)	+
UF7	**1.1844e + 0 (6.98e-2)**	1.4398e + 0 (7.08e-2)	+	1.4175e + 0 (1.54e-1)	+	1.3932e + 0 (7.04e-2)	+	1.4151e + 0 (5.15e-2)	+	1.4735e + 0 (8.20e-2)	+
ZDT1	**2.1458e + 0 (1.61e-1)**	2.2803e + 0 (8.35e-2)	+	2.3207e + 0 (7.92e-2)	+	2.2703e + 0 (9.71e-2)	=	2.2398e + 0 (1.03e-1)	=	2.2124e + 0 (1.29e-1)	=
ZDT2	**3.1798e + 0 (1.52e-1)**	3.6076e + 0 (8.90e-2)	+	3.5827e + 0 (1.47e-1)	+	3.6566e + 0 (1.44e-1)	+	3.5687e + 0 (1.57e-1)	+	3.6034e + 0 (1.30e-1)	+
ZDT3	1.8740e + 0 (1.86e-1)	1.8278e + 0 (1.26e-1)	=	1.8643e + 0 (9.65e-2)	=	1.8484e + 0 (1.23e-1)	=	**1.8094e + 0 (1.61e-1)**	=	1.8573e + 0 (1.56e-1)	=
ZDT4	**5.7361e + 2 (2.31e + 1)**	6.7089e + 2 (1.88e + 1)	+	6.7011e + 2 (1.61e + 1)	+	6.5855e + 2 (3.09e + 1)	+	6.5216e + 2 (2.43e + 1)	+	6.5904e + 2 (2.54e + 1)	+
ZDT6	**7.2155e + 0 (9.42e-2)**	7.3656e + 0 (4.23e-2)	+	7.3830e + 0 (6.13e-2)	+	7.4104e + 0 (6.31e-2)	+	7.3773e + 0 (8.10e-2)	+	7.3689e + 0 (7.98e-2)	+
DTLZ1	**1.6110e + 2 (4.79e + 1)**	1.4256e + 3 (6.63e + 1)	+	1.4282e + 3 (5.57e + 1)	+	1.4430e + 3 (5.44e + 1)	+	1.4214e + 3 (5.27e + 1)	+	1.4006e + 3 (8.69e + 1)	+
DTLZ2	**1.5913e + 0 (1.04e-1)**	2.7170e + 0 (1.13e-1)	+	2.6496e + 0 (1.62e-1)	+	2.6362e + 0 (1.05e-1)	+	2.6890e + 0 (1.18e-1)	+	2.6919e + 0 (1.77e-1)	+
DTLZ3	**1.4848e + 2 (1.30e + 2)**	3.7678e + 3 (1.46e + 2)	+	3.8462e + 3 (1.35e + 2)	+	3.8790e + 3 (1.06e + 2)	+	3.7994e + 3 (2.33e + 2)	+	3.7938e + 3 (2.50e + 2)	+
DTLZ4	**1.9301e + 0 (1.21e-1)**	3.0040e + 0 (1.44e-1)	+	3.0341e + 0 (1.39e-1)	+	3.0328e + 0 (1.38e-1)	+	3.0445e + 0 (1.15e-1)	+	3.0416e + 0 (1.15e-1)	+
+/=/-	**NA**	**14/1/1**		**14/1/1**		**13/2/1**		**13/2/1**		**13/2/1**	

various dimensions. We speculate that it may be SaMOPSO_NS is not adept at dealing with UF4 given the distinctive properties of this issue. After all, no algorithm, no matter how sophisticated, can solve all issues well. For ZDT3, as previously stated, it is a highly complicated MOP with discontinuous features that most SAEAs struggle to solve. Generally speaking, based on the test instances examined in this study, it can be inferred that SaMOPSO_NS indeed holds a strong competitive stance among the evaluated algorithms.

4.5. Expensive EMAT optimization

a) Problem Description.

In this subsection, we implement our proposed SaMOPSO_NS to tackle real-world performance index optimization problem found in multi-objective electromagnetic acoustic transducers [43]. This involves a typical expensive MOP where the precise mathematical formula of the objective function remains elusive, and the objective value can only be procured through time-consuming simulation in COMSOL software.

In performance indices optimization of EMAT, our purpose aims to improve the signal amplitude ($A_{u,i(x)}$) at z-axial, designated as the first objective f_1, as well as to decrease the amplitude ratio ($r_{i(x)}$) at ith sample point x, defined as the second objective f_2. This expensive MOP can be described informally as follows:

$$\begin{aligned} &\min imizeF(x) = \{f_1(x), f_2(x)\} \\ &s.tf_1(x) = \Phi_1(x) = -A_{u,i(x)} \\ &\quad f_2(x) = \Phi_2(x) = r_{i(x)} \\ &\quad x = [R_{m1}, R_{m2}, H_m, g_c, I] \end{aligned} \tag{8}$$

where Φ_1 and Φ_2 symbolize the unknown relationship between the objectives and the decision variables within performance indices optimization of EMAT. The sample point, denoted as x, consists of five important decision variables. R_{m1} represents the radius of the cylindrical magnet and $R_{m1}\in$[1,5]mm. R_{m2} represents the inner diameter of the ring magnet and $R_{m2}\in$[10,12]mm. H_m represents the height of the magnet and $H_m\in$[10,20]mm. g_c represents the lift-off distance and $g_c \in$ [0.1, 1]mm. I represents the exciting current and $I \in$ [50,240]A. In COMSOL software, for a sample point x, it takes about 25 min to run a simulation analysis. Fig. 6 displays the physical structure of the designed EMAT.

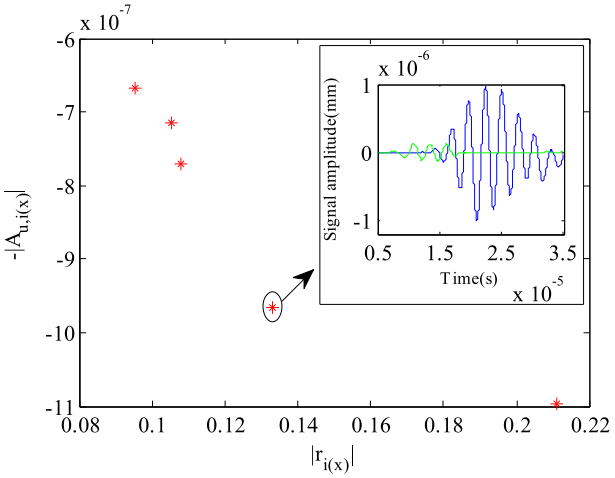


Fig. 6. Geometrical parameters of the Lamb waves EMAT.

Table 7
The experimental results of the designed EMAT.

R_{m1}	R_{m2}	H_m	g_c	I	$(-A_{u,i(x)}, r_{i(x)})$
3.5820	12.0563	12.4210	0.1093	223.1167	(-1.0970E-06, 0.2113)
3.2598	13.7952	17.3965	0.1097	217.9001	(-9.6660E-07, 0.1328)
2.6991	14.0798	11.1982	0.1426	215.3611	(-7.7033E-07, 0.1075)
2.5176	16.6462	11.7816	0.1481	233.4039	(-7.1453E-07, 0.1050)
2.2363	16.3482	12.2622	0.1098	221.1973	(-6.6694E-07, 0.0952)

b) Optimization Result.

In this specific case, it is important to note that the proposed algorithm requires slight modification due to the inability to calculate the IGD value. In the modified algorithm, the EXA-NS strategy continues to be performed periodically with $NS\_int = 2$, without the need to determine whether the IGD condition has been met. This is because it is not possible to obtain the true PF of the practical expensive MOP, particularly for black-box problems. In this study, the maximum expensive FEs is set to 300 and $NP = 20$.

Table 7 presents the experimental results, clearly illustrating five distinct solutions (marked with *) on the Pareto front are not closely located to each other. If we do not have any additional information to aid our decision-making, the Pareto front serves as the final solution for this multi-objective optimization issue. Considering the special structure of this particular Pareto front, one viable and sensible option is to select the knee solution, represented by the solution in the black circle in Fig. 7. That is, the second solution in Table 7 corresponds to (3.2598, 13.7952, 17.3965, 0.1097, 217.9001) where both objectives reach their compromise minimum. At the knee solution, the time waveforms and objective values in the forward direction are depicted in the subfigure of Fig. 7. Based on the results, the optimized EMAT designed in this paper achieves an amplitude 9.666E-07 mm and an amplitude ratio 0.13, which significantly outperforms the reported results [43]. In practice, a higher amplitude value indicates a stronger signal and implies a higher detection sensitivity of the transducer product. Conversely, a smaller amplitude ratio means less noise, which facilitates the accurate transmission of signals and readings.

5. Conclusion

This paper introduces an ensemble surrogate-assisted MOPSO algorithm to efficiently solve medium-dimensional expensive MOPs within hundreds of function evaluations. The algorithm incorporates a *pbest*-dominated infill criterion and an external archive-based neighborhood search strategy to strike an optimal balance between exploration and exploitation. The infill criterion is employed to evaluate prospective candidates realistically, facilitating an exploration of overall regions with well-distributed individuals. Conversely, the neighborhood search strategy is used as a localized search to exploit current promising domains meticulously. To evaluate the efficiency of the designed SaMOPSO_NS algorithm, it is applied to three different types of widely-used multi-objective benchmark functions and an engineering multi-objective EMAT optimization problem. The extensive experimental results suggest that the proposed SaMOPSO_NS outperforms the compared algorithm in terms of convergence speed and accuracy for expensive MOPs in medium dimensions. The optimized single-point amplitude and amplitude ratio of the EMAT, designed in this paper, reach 9.666E-07 mm and 0.1328 respectively, thus achieving higher sensitivity.

However, for low-dimensional issues and discontinuous problems, results show that the proposed algorithm performs poorly compared to other competing algorithms. Therefore, it deserves further efforts to improve SaMOPOS_NS' ability to address these

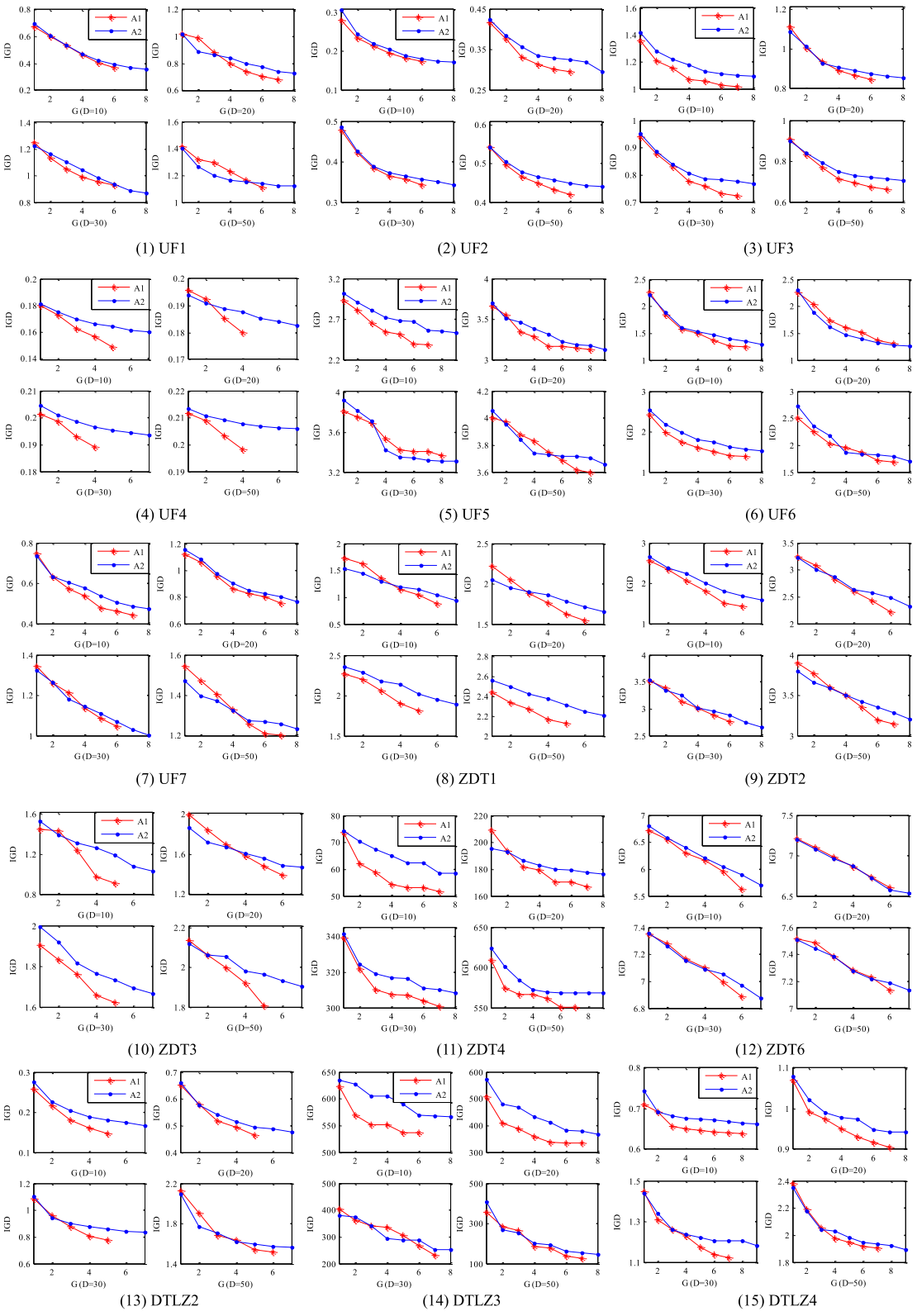


Fig. 7. The obtained Pareto front.

expensive MOPs with irregular PFs [44], such as unconnected, degenerate, non-convex/concave [45] and others in the follow-up works.

CRediT authorship contribution statement

Mingyuan Yu: Writing – review & editing, Writing – original draft, Methodology, Investigation, Formal analysis, Data curation. **Zhou Wu:** Writing – review & editing, Supervision, Methodology, Funding acquisition. **Jing Liang:** Writing – review & editing, Visualization, Supervision, Methodology, Formal analysis. **Caitong Yue:** Writing – review & editing, Visualization, Validation, Software, Methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (2021YFF0500903), the National Natural Science Foundation of China (Grant No. 62306285), the China Postdoctoral Science Foundation (Grant No. 2023TQ0318, Grant No. GZC20232397), the Key Laboratory of Big Data Intelligent Computing, Chongqing University of Posts and Telecommunications (Grant No. BDIC-2023-A-007), and to which the authors are very grateful.

References

[1] Y. Jin, B. Sendhoff, A systems approach to evolutionary multiobjective structural optimization and beyond, IEEE Computational Intelligence Magazine. 4 (3) (2009) 62–76.
[2] S.K. Goudos, K.A. Gotsis, K. Siakavara, et al., A multi-objective approach to subarrayed linear antenna arrays design based on memetic differential evolution, IEEE Transactions on Antennas and Propagation. 61 (6) (2013) 3042–3052.
[3] H. Wang, Y. Jin, A random forest-assisted evolutionary algorithm for data-driven constrained multiobjective combinatorial optimization of trauma systems, IEEE Transaction on Cybernetics. 50 (2) (2020) 536–549.
[4] D. Zhan, Y. Cheng, J. Liu, Expected improvement matrix-based infill criteriafor expensive multiobjective optimization, IEEE Transactions on Evolutionary Computation. 21 (6) (2017) 956–975.
[5] Z. Wang, Q. Zhang, Y.-S. Ong, et al., "Choose appropriate subproblems for collaborative modeling in expensive multiobjective optimization", IEEE Transactions on Cybernetics. 1-14, 2021.
[6] R. Jiao, S. Zeng, C. Li, et al., A complete expected improvement criterion for gaussian process assisted highly constrained expensive optimization, Information Sciences. 80–96 (2019).
[7] D.R. Jones, A taxonomy of global optimization methods based on response surfaces, Journal of Global Optimization. 21 (4) (2001) 345–383.
[8] M. Yu, J. Liang, K. Zhao, et al., An aRBF surrogate-assisted neighborhood field optimizer for expensive problems, Swarm and Evolutionary Computation. 68 (2021) 100972.
[9] H. Dong, J. Li, P. Wang, et al., Surrogate-guided multi-objective optimization (SGMOO) using an efficient online sampling strategy, Knowledge-Based Systems. 220 (2021) 106919.
[10] Z. Song, H. Wang, C. He, et al., A kriging-assisted two-archive evolutionary algorithm for expensive many-objective optimization, IEEE Transactions on Evolutionary Computation. 25 (6) (2021) 1013–1027.
[11] P. Liao, C. Sun, G. Zhang, et al., Multi-surrogate multi-tasking optimization of expensive problems, Knowledge-Based Systems. 205 (2020) 106262.
[12] S.C. Chu, Z.G. Du, Y.J. Peng, et al., Fuzzy hierarchical surrogate assists probabilistic particle swarm optimization for expensive high dimensional problem, Knowledge-Based Systems. 220 (2021) 106939.
[13] J. Liang, H. Lin, C. Yue, et al., Multiobjective differential evolution with speciation for constrained multimodal multiobjective optimization, IEEE Transactions on Evolutionary Computation. 27 (4) (2023) 1115–1129.
[14] C. Yang, J. Ding, Y. Jin, et al., Offline data-driven multiobjective optimization knowledge transfer between surrogates and generation of final solutions, IEEE Transactions on Evolutionary Computation. 24 (3) (2020) 409–423.
[15] X.L. Wang, Y.C. Jin, S. Schmitt, et al., An adaptive bayesian approach to surrogate-assisted evolutionary multi-objective optimization, Information Sciences. 519 (2020) 317–331.
[16] X. Wang, Y. Jin, S. Schmitt, et al., Transfer learning based surrogate assisted evolutionary bi-objective optimization for objectives with different evaluation times, Knowledge-Based Systems. 227 (2021) 107190.
[17] Z. Chen, Y. Zhou, X. He, Handling expensive multi-objective optimization problems with a cluster-based neighborhood regression model, Applied Soft Computing. 80 (2019) 211–225.
[18] F. Li, L. Gao, A. Garg, et al., Two infill criteria driven surrogate-assisted multi-objective evolutionary algorithms for computationally expensive problems with medium dimensions, Swarm and Evolutionary Computation. 60 (2021) 100774.
[19] X. Xiang, Y. Tian, J. Xiao, et al., "A Clustering-Based Surrogate-Assisted Multiobjective Evolutionary Algorithm for Shelter Location Problem Under Uncertainty of Road Networks", IEEE transection on Industrial Informatics. 2020.
[20] D. Guo, Y. Jin, J. Ding, et al., Heterogeneous ensemble-based infill criterion for evolutionary multiobjective optimization of expensive problems, IEEE Transaction on Cybernetics. 49 (3) (2019) 1012–1025.
[21] Z. Lv, L. Wang, Z. Han, et al., Surrogate-assisted particle swarm optimization algorithm with pareto active learning for expensive multi-objective optimization, IEEE CAA Journal of Automatica Sinica. 6 (3) (2019) 838–849.
[22] Z. Liu, H. Wang, A data augmentation based kriging-assisted reference vector guided evolutionary algorithm for expensive dynamic multi-objective optimization, Swarm and Evolutionary Computation. 75 (2022) 101173.

[23] T. Chugh, Y. Jin, K. Miettinen, et al., A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization, IEEE Transactions on Evolutionary Computation. 22 (1) (2018) 129–142.
[24] P. Jiang, Y. Cheng, J. Yi, et al., An efficient constrained global optimization algorithm with a clustering-assisted multiobjective infill criterion using gaussian process regression for expensive problems, Information Sciences. 569 (2021) 728–745.
[25] L. Pan, C. He, Y. Tian, et al., A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization, IEEE Transactions on Evolutionary Computation. 23 (1) (2019) 74–88.
[26] H. Jie, Y. Wu, J. Zhao, et al., An efficient multi-objective PSO algorithm assisted by kriging metamodel for expensive black-box problems, Journal of Global Optimization. 67 (1–2) (2016) 399–423.
[27] L. Tang, X. Wang, A hybrid multiobjective evolutionary algorithm for multiobjective optimization problems, IEEE Transactions on Evolutionary Computation. 17 (1) (2013) 20–45.
[28] M. Yu, J. Liang, Z. Wu, et al., A twofold infill criterion-driven heterogeneous ensemble surrogate-assisted evolutionary algorithm for computationally expensive problems, Knowledge-Based Systems. 236 (2022) 107747.
[29] Z. Wu, Neighborhood field for cooperative optimization, Soft Computing. 17 (5) (2013) 819–834.
[30] Z. Wu, T.W.S. Chow, A local multiobjective optimization algorithm using neighborhood field, Structural and Multidisciplinary Optimization. 46 (46) (2012) 853–870.
[31] F.A.C. Viana, G. Venter, V. Balabanov, An algorithm for fast optimal latin hypercube design of experiments, International Journal for Numerical Methods in Engineering. 82 (2) (2010) 135–156.
[32] K. Deb, A. Pratap, S. Agarwal, et al., A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation. 6 (2) (2002) 182–197.
[33] P.A.N. Bosman, D. Thierens, The balance between proximity and diversity in multiobjective evolutionary algorithms, IEEE Transactions on Evolutionary Computation. 7 (2) (2003) 174–188.
[34] H. Wang, Y. Jin, C. Sun, et al., Offline data-driven evolutionary optimization using selective surrogate ensembles, IEEE Transactions on Evolutionary Computation. 23 (2) (2019) 203–216.
[35] L. Breiman, Bagging predictors, Machine Learning. 24 (2) (1996) 123–140.
[36] Y. Bi, B. Xue, M. Zhang, Using a small number of training instances in genetic programming for face image classification, Information Sciences. 593 (2022) 488–504.
[37] M. Yu, X. Li, J. Liang, A dynamic surrogate-assisted evolutionary algorithm framework for expensive structural optimization, Structural and Multidisciplinary Optimization. 61 (2) (2020) 711–729.
[38] N. Garcia-Pedrajas, C. Hervas-Martinez, D. Ortiz-Boyer, Cooperative coevolution of artificial neural network ensembles for pattern classification, IEEE Transactions on Evolutionary Computation. 9 (3) (2005) 271–302.
[39] J. Knowles, ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems, IEEE Transactions on Evolutionary Computation. 10 (1) (2006) 50–66.
[40] Q. Zhang, W. Liu, E. Tsang, et al., Expensive multiobjective optimization by MOEA/D with gaussian process model, IEEE Transactions on Evolutionary Computation. 14 (3) (2010) 456–474.
[41] J. Luo, A. Gupta, Y.S. Ong, et al., Evolutionary optimization of expensive multiobjective problems with co-sub-pareto front gaussian process surrogates, IEEE Transactions on Cybernetics. 49 (5) (2019) 1708–1721.
[42] Y. Tian, R. Cheng, X. Zhang, et al., PlatEMO: a MATLAB platform for evolutionary multi-objective optimization, IEEE Computational Intelligence Magazine. 12 (4) (2017) 73–87.
[43] S. Wang, S. Huang, A. Velichko, et al., A multi-objective structural optimization of an omnidirectional electromagnetic acoustic transducer, Ultrasonics. 81 (2017) 23–31.
[44] C. Yue, J. Liang, B. Qu, et al., A survey on multimodal multiobjective optimization, Control and Decision. 36 (11) (2021) 2577–2588.
[45] J. Liang, B. Y. Qu, B. L. Li, et al., "Locating multiple roots of nonlinear equation systems via multi-strategy optimization algorithm with sequence quadratic program", Science China-Information Sciences. 65(7), 2022.