Infill Criterion Ensemble in Multi-Objective Evolutionary Algorithm for Mixed-Variable Problems

1st Yongcun Liu
School of Artificial Intelligence
Xidian University
Xi'an, China
yongcunl@stu.xidian.edu.cn

2nd Handing Wang
School of Artificial Intelligence
Xidian University
Xi'an, China
hdwang@xidian.edu.cn

3rd Jingjing Ma
School of Artificial Intelligence
Xidian University
Xi'an, China
jjma@xidian.edu.cn

***Abstract*—Many real-world optimization problems involve mixed variables, multiple conflicting objectives, and computationally expensive evaluations. Such problems are called expensive mixed-variable multi-objective optimization problems (EMVMOPs). Solving EMVMOPs is challenging due to a complex search space involving mixed variables, balancing conflicts among multiple objectives, and a limited number of function evaluations. In this work, we propose an infill criterion ensemble in surrogate-assisted multi-objective evolutionary algorithm, which primarily consists of two core components considering the convergence to non-dominated front, model uncertainty, and diversity. We use two indicators, ensemble through an adaptive weighted sum, to play a crucial role in enhancing diversity, while local search with hybrid operators contributes to improving local convergence. The experimental results show that our algorithm is competitive compared to other algorithms on benchmark problems.**

***Index Terms*—Mixed-variable, multi-objective, infill criterion ensemble, surrogate-assisted evolutionary algorithm, local search, hybrid operators**

I. Introduction

Many real-world optimization problems require to simultaneously optimize multiple conflicting objectives, and the decision variables are of mixed types, including both continuous and discrete variables [1]. In such problems there are no explicit expressions, and computationally expensive, which implies a substantial consumption of computational resources and time [2]. Examples of such problems come from optimization of wind farm layout [1], ordinary differential equation catalytic system and fuel injector design [3], etc. The aforementioned category of problems can be defined as the expensive mixed-variable multi-objective optimization problems (EMVMOPs), which can be formulated as follows:

$$\begin{aligned} &\min_{\boldsymbol{x},\boldsymbol{z}} F(\boldsymbol{x},\boldsymbol{z}) = \{f_1(\boldsymbol{x},\boldsymbol{z}), f_2(\boldsymbol{x},\boldsymbol{z}), ..., f_M(\boldsymbol{x},\boldsymbol{z})\} \\ &\quad s.t.\ \boldsymbol{x} \in \mathcal{F}_{\boldsymbol{x}} \subseteq \mathbb{R}^{n_1},\ \boldsymbol{z} \in \mathcal{F}_{\boldsymbol{z}} \end{aligned} \tag{1}$$

where M is the number of objectives, $\boldsymbol{x}$ is a n_1-dimensional decision vector in the continuous space $\mathcal{F}_{\boldsymbol{x}}$ and $\boldsymbol{z}$ is a discrete decision vector in the discrete space $\mathcal{F}_{\boldsymbol{z}}$. The objective functions $f_i(\boldsymbol{x},\boldsymbol{z}), i = 1, .., M$ are computationally expensive. The main challenges in addressing EMVMOPs arise from the search space with mixed variables, conflicts among multiple objectives, and the limited function evaluations.

Typically, effective handling of mixed variables plays a crucial role in evolutionary operators [4] and surrogate models [5]. In recent years, the methods for handling mixed variables can be roughly categorized into three types: 1) discretization method [6]: discretizing continuous variables and optimizing using a discrete optimizer; 2) relaxation method [7]: relaxing continuous variables and decoding during evaluation; 3) two-partition method [8]: employing different operators to separately search in continuous and discrete spaces. In general, the first method entails a loss of optimization precision, the second involves quantization errors, and the third may compromise the consistency of optimization.

To balance multiple objectives under a finite number of function evaluations, there has been a series of relevant work [2] in recent years. According to the optimization approach, it can be classified into three categories: 1) Pareto-dominance based [9], [10]; 2) Decomposition based [11], [12]; 3) Indicator based [13], [14]. Besides, some infill criteria, such as Expected Improvement (EI), Probability of Improvement (PI), and Upper Confidence Bound (UCB) [15] have been proposed for single-objective optimization problems. For expensive multi-objective optimization problems (EMOPs), in addition to the optimizer, designing appropriate infill criteria is also crucial. In solving EMOPs, infill sampling is typically based on convergence, model uncertainty, and diversity to select promising solutions from a set of candidate solutions for function evaluation, updating the evaluated solution set and the surrogate model. Because we need to rely on the surrogate model to guide the population towards Pareto Front (PF), and considering both the approximation error of the surrogate model and the diversity of solutions is also essential. Recent work has proposed some algorithms that perform reference vector-guided surrogate-assisted multi-objective optimization firstly, followed by infill sampling based on indicators, such

This work was supported in part by the National Natural Science Foundation of China (No. 62376202).(*Corresponding author: Handing Wang*)

as PB-NSGAIII [16], EMMOEA [17]. The first algorithm employs a surrogate-assisted multi-objective optimization algorithm (NSGAIII [18] or RVEA [19]) to obtain a set of well-converged and diverse trial solutions, and re-evaluates non-dominated solutions based on two indicators that measure convergence and diversity as optimization objectives. The second utilizes a surrogate-assisted reference vector-guided search to identify promising candidate solutions, and the solution among the identified candidates with the maximum EI in terms of the performance indicator is re-evaluated. Both algorithms are effective on EMOPs. However, the first approach lacks consideration for model uncertainty during infill sampling, while the second one requires additional training of model approximation indicators, resulting in increased complexity. More importantly, due to the lack of consideration for mixed variables, they are struggle to achieve good performance on EMVMOPs.

Recently, there have been some algorithms proposed specifically to address EMVMOPs. Manson et al. [3] proposed a mixed-variable multi-objective optimization based on the expected improvement matrix (EIM) based infill sampling criterion and Gower similarity-based Gaussian processes regression (GPR). Sun et al. [20] proposed an algorithm that incorporates several evolutionary operators to handle different types of decision variables and utilizes adaptive sampling based on approximate objective values and diversity. They employ hybrid operators to handle mixed variables, but lack consideration of model uncertainty during sampling. Sheikh et al. [15] proposed a modified Hedge strategy that utilizes a portfolio of acquisition functions for EMVMOPs. In general, existing algorithms lack a comprehensive consideration of the convergence, diversity, and model uncertainty of solutions during infill sampling.

To comprehensively consider the convergence, diversity, model uncertainty, and mixed variables during the optimization process, we propose a infill criterion ensemble in surrogate-assisted multi-objective evolutionary algorithm (ICE-MOEAmv) to solve EMVMOPs. Our proposed algorithm primarily consists of two core components: infill criterion ensemble-based sampling and local search using hybrid operators. The main contributions of this algorithm as follows:

- We propose an infill criterion ensemble strategy, which involves weighting the EIM value and the distance indicator (DI) based on the proportion of non-dominated solutions in the current archive. The EIM provides convergence and uncertainty in multiple optimization directions, while DI considers diversity. Combining these two indicators allows for a more comprehensive global infill sampling.
- Local search based on hybrid operators is proposed to enhance convergence. In addition, handling mixed variables involves using different operators to mutate the continuous and discrete variables separately, which further searches in the local space.

The remainder of this paper is arranged as follows: Reference vector guided evolutionary algorithm, indicator-based evolutionary algorithm and EIM-based infill criteria are introduced in Section II. The details of our proposed algorithm are presented in Section III. The experimental results and the analysis are shown in the Section IV. Finally, the conclusions and the future work are given in Section V.

II. Related Work

A. Reference Vector Guided Search

In a reference vector-guided evolutionary algorithm for multi-objective optimization problems, a set of evenly distributed reference vectors is generated to obtain representative solutions of the entire PF [19]. Based on the reference vectors, the diversity of solutions is reasonably ensured. Combined with angle-penalized distance for solution selection, convergence can be assured in many-objective optimization problems. However, in surrogate-assisted multi-objective optimization, the use of aggregation functions can lead to accumulated errors due to the approximation errors in the surrogate model. The Euclidean distance to the ideal point is also can be used for environmental selection, as described in [17], [21]. Firstly, the reference vectors $\{V_1, V_2, ..., V_n\}$ are generated using Das and Dennis' systematic approach [18]. A surrogate model is built for each objective. The population is initialized, and individuals are evaluated using the trained surrogate models. The offspring is generated through crossover and mutation of the population, evaluated using the surrogate model, and then merged with the population. The objective predictions $\{\hat{f}_1, \hat{f}_1, ..., \hat{f}_M\}$ of the merged population are normalized. As shown in Fig.1, the main steps of the reference vector-guided

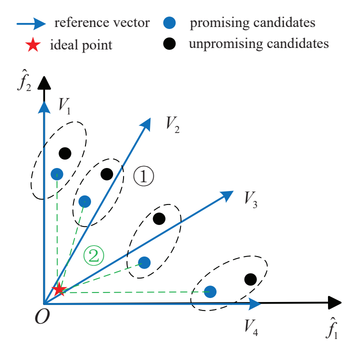

Fig. 1. Example to show environmental selection

environmental selection are as follows:

1) The individuals in the merged population are assigned to their nearest reference vector based on the normalized objective predictions.
2) For each reference vector associated with at least one solution, the candidates closest to the ideal point (promising candidates) are selected for the next generation.

B. Indicator-Based Selection

The main idea of the indicator-based selection [13] is to formalize preferences in terms of continuous generalizations of the dominance relation, which does not require any additional diversity preservation mechanism. Meanwhile, local search is often used to strengthen the exploitation, which does not need to exert too much effort on maintaining diversity. Therefore, using indicator-based selection in local search can enhance the convergence of the algorithm [22]. The binary additive ϵ-indicator $I_{\epsilon+}$ can be used to compare the quality of two Pareto sets $\{A,\ B\}$ approximations relatively to each other, its formula is as follows:

$$I_{\varepsilon+}(A,B)=\min_{\varepsilon}\{\forall \boldsymbol{x_2}\in B,\exists \boldsymbol{x_1}\in A: \\ f_i(\boldsymbol{x_1})-\varepsilon\leqslant f_i(\boldsymbol{x_2}),\forall i\in\{1,...,M\}\} \tag{2}$$

Based on the $I_{\epsilon+}$, the fitness value of solution $\boldsymbol{x_1}$ in population P can be calculated using the following formula:

$$F(\boldsymbol{x_1})=\sum_{\boldsymbol{x_2}\in P\backslash\{\boldsymbol{x_1}\}} -e^{-I_{\epsilon+}(\{\boldsymbol{x_2}\},\{\boldsymbol{x_1}\})/0.05} \tag{3}$$

The pseudocode for environmental selection based on the fitness definition mentioned above is shown in Algorithm 1: we can select K promising solutions from the candidates based

Algorithm 1 Indicator-based selection

Input: P: candidates, K: the number of selected solutions;
Output: P_K: selected solutions;
Calculate the fitness values for all individuals in P using Eq.(3), $t=0$;
while $t<K$ **do**
  Choose an individual $\boldsymbol{x}^*$ with the smallest fitness value from P;
  Remove $\boldsymbol{x}^*$ from P and update the fitness values of the remaining individuals using Eq.(3), $t=t+1$;
end while
$P_K\leftarrow P$;

on $I_{\epsilon+}$, which is suitable for local search.

C. Expected Improvement Matrix-Based Infill Criteria

Ultimately, we need surrogate models to approximate multiple objectives, replacing a large number of expensive function evaluations in the evolutionary optimization process [23]. However, the approximation errors generated by the surrogate model predictions are unavoidable, and the model uncertainty needs to be considered when infill sampling. The GPR model is a common surrogate model capable of predicting the distribution of objectives [10]. It provides information about the uncertainty of the predictions. For a well-trained GPR model, the prediction of the j-th objective on solution x is as follows:

$$\hat{y}_j(\boldsymbol{x})\sim N(\hat{f}_j(\boldsymbol{x}),s_j^2(\boldsymbol{x})) \tag{4}$$

where $\hat{f}_j(\boldsymbol{x})$ is prediction mean, $s_j(\boldsymbol{x})$ is standard deviation, and $N(\cdot)$ represents a Gaussian distribution. For single-objective optimization, the EI criterion is commonly used as the infill criteria, which guides optimization towards the direction with the maximum expected improvement, taking the current best solution as a reference. In EIM infill criteria [24] for EMOPs, the current non-dominated front $\{f_j^i, i=1,...,N_{nds}, j=1,...M\}$ with N_{nds} solutions become the reference. Then EI can be calculated as follows:

$$EI(\hat{f}_j,f_j^i)=(f_j^i-\hat{f}_j(\boldsymbol{x}))\Phi(\frac{f_j^i-\hat{f}_j(\boldsymbol{x})}{s_j(\boldsymbol{x})})+ \\ s_j(\boldsymbol{x})\phi(\frac{f_j^i-\hat{f}_j(\boldsymbol{x})}{s_j(\boldsymbol{x})}) \tag{5}$$

where Φ and ϕ are the Gaussian cumulative distribution function and probability density function, respectively. Using Eq.5, we can obtain an $N_{nds}\times M$ EI matrix. Based on this matrix, the EIM value for each solution can be calculated using the following formula:

$$EIM(\boldsymbol{x})=\min_i\sqrt{\sum_{j=1}^{M}EI(\hat{f}_j(\boldsymbol{x}),f_j^i)^2} \tag{6}$$

Eq.6 indicates that the EIM value is obtained by first calculating the root mean square on the objective dimension and then optimizing in the direction of the current non-dominated solution with the smallest EI.

III. Proposed Algorithm

The reference vector guided search can maintain diversity during the optimization process, and the simple selection rule based on the Euclidean distance to the ideal point does not lead to much cumulative prediction errors. The EIM indicator can guide optimization towards the direction of non-dominated solutions with the maximum optimization potential. The indicator-based selection is suitable for local search. In order to better to solve EMVMOPs, we propose an infill criterion ensemble in surrogate-assisted multi-objective evolutionary algorithm. The flowchart of our proposed algorithm is shown in Fig.2. The main steps of the algorithm are as follows:

1) **Initialization**: firstly, the algorithm generates an initial archive using Latin hypercube sampling method [25] and adds them into database after performing expensive function evaluations.
2) **Global reference vector-guided search**: as described in Algorithm 2, the global GPR model M_g is trained using the database DB , and the current archive $Arch$ serves as the initial population (lines 1-2). The following GPR-assisted reference vector-based search (lines 4-10) will be repeated for w_{max} iterations.
3) **Infill criterion ensemble-based sampling**: the EIM and DI are integrated through a weighted sum approach, and the solution with maximum integrated indicator is selected from the population for re-evaluation.
4) **Local search using hybrid operators**: the two-partition operators are applied to the current non-dominated solutions, and indicator-based selection is used to choose the top K promising solutions for re-evaluation.
5) **Termination condition**: when the maximum number of expensive function evaluations is exhausted, the non-dominated solutions will be the output. Otherwise, the algorithm proceeds to step 2) next.

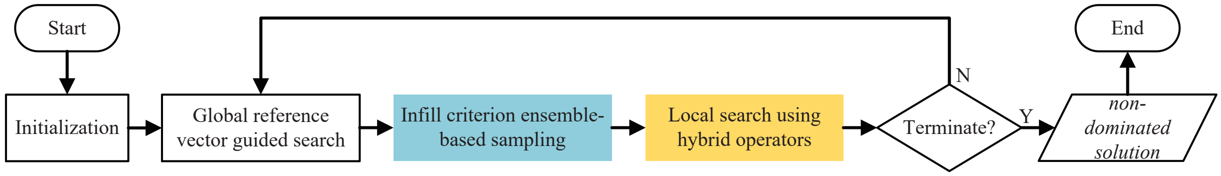

Fig. 2. Flowchart of our proposed ICE-MOEAmv

Algorithm 2 Global reference vector-guided search

Input:
RV: a set of given reference vectors;
$Arch$: current evaluated archive, size of N;
DB: current evaluated solution database;
Output:
P: the last population;

```
Utilizing the database, build or update GPR model M_g as surrogate model for each objective;
Employing Arch for population P initialization;
/* Surrogate-assisted reference vector guided search */
while t ≤ w_max do
    Generate offspring population O by performing crossover and mutation on the P;
    Predict the objective values of solutions in O using M_g;
    P' ← ∅ and PO ← P ∪ O;
    Min-max normalize the approximated objective values, Assign each solution in PO to its closest reference vector;
    while |P'| ≤ N do
        P' add the solution s closest to the ideal point within all solutions assigned to the reference vector;
        Remove the solution s from the reference vector;
    end while
    P ← P' and t = t + 1;
end while
```

To ensure the continuity and consistency of the genetic operators in the global reference vector-guided search, a relaxation method is employed to handle mixed variables. The specific operators include simulated binary crossover and polynomial mutation, with rounding to the nearest integer applied to the discrete part. Two partition method is employed in local search for handling mixed variables and local convergence. In the subsequent subsections, we will elaborate on the details of the two core components of our algorithm.

A. Infill Criterion Ensemble-Based Sampling

To simultaneously consider convergence, diversity, and model uncertainty during infill sampling, the EIM value and DI are ensembled through a weighted sum. Note that a fixed number of N solutions are selected as candidates P in the global reference vector-guided search and the ideal point is $F^* = \{f_1^{min}, f_2^{min}, ..., f_M^{min}\}, f_j^{min} < f_j^i, i \in Arch$. After obtaining the output population P, the EIM values for each solution are calculated using Eq.(6). The formula of DI is as follows:

$$DI(\boldsymbol{x}) = \min_i ||\hat{F}(\boldsymbol{x}) - F^i||_2 \tag{7}$$

where $F^i = \{f_1^i, ..., f_M^i\}$ is objective values of the i-th non-dominated solution in $Arch$. Then, the EIM value and DI are weighted using the following formula:

$$indicator(\boldsymbol{x}) = (1-c) \times EIM(\boldsymbol{x}) + c \times DI(\boldsymbol{x}) \tag{8}$$

where c represents the proportion of non-dominated solutions in $Arch$. This ensemble method can adaptively enhance convergence when the number of non-dominated solutions is small and promote diversity when there are many non-dominated solutions. At last, the best candidate with the maximum ensemble indicator is selected for re-evaluation from the last population P and added to the DB, then $Arch$ will be updated.

B. Local Search Using Hybrid Operators

In the local search module, we design a two-partition method to handle mixed variables, and utilize $I_{\epsilon+}$-based environmental selection to enhance local convergence. The flowchart of local search is illustrated in Fig.3. Initially, non-

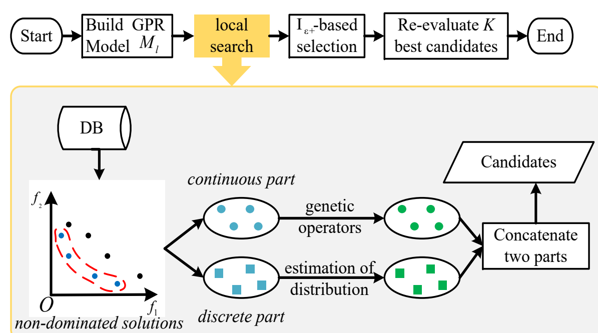

Fig. 3. Flowchart of local search using hybrid operators

dominated solutions are selected, and decision variables are divided into two parts: continuous and discrete parts. Note that when there are fewer than ten non-dominated solutions, the first ten solutions are selected. Simulated binary crossover and polynomial mutation (genetic operators) are applied to the continuous part, while sampling is performed on the discrete part based on selection probability (estimation of distribution). The weight formula for discrete value selection is as follows:

$$w_{j,k} = \begin{cases} \frac{1}{\text{dim}_d \times {n_j}^{M-1}}, if\ ct_{j,k} == 0 \\ \frac{ctb_{j,k}}{ct_{j,k}} + \frac{1}{\text{dim}_d \times {n_j}^{M-1}}, otherwise \end{cases} \tag{9}$$

where $w_{j,k}$ represents the weight for selecting the k-th discrete value for the j-th discrete variable. dim_d and n_j respectively denote the number of discrete variables and the number of possible values for the j-th discrete variable. $ct_{j,k}$ is the count of samples in the $Arch$ where the j-th discrete variable takes the k-th discrete value, and $ctb_{j,k}$ is the count of samples in the first $N_{ctb} = min(N_{nds}, 0.45 * N)$ individuals where the j-th discrete variable takes the k-th discrete value. Next, the selection probability can be calculated using the following formula:

$$prob_{j,k} = \frac{w_{j,k}}{\sum\limits_{k} w_{j,k}}. \tag{10}$$

The continuous and discrete parts of the offspring generated by the hybrid operators are concatenated. Subsequently, the GPR model M_l trained on $Arch$ is employed to predict the objective values. $I_{\epsilon+}$-based environmental selection, as depicted in Algorithm 1, is then utilized to select K best solutions for re-evaluation. Finally, the re-evaluated solutions are added to the DB, and $Arch$ will be updated.

IV. Experiments

A. Test Problems and Experiment Settings

Since most benchmarks are designed for continuous problems, to assess the performance of our proposed algorithm on the EMVMOPs, we modified existing ZDT and DTLZ problems, as mentioned in this paper [26]. To enhance the realism and complexity of the benchmark, discrete variables have two modes of value assignment: uniform distribution and non-uniform distribution. Furthermore, the problem involves discrete variables at different ratios, such as 0.2, 0.5, 0.8. The discrete variables will include the position parameters of the problem, signifying that the PF will be influenced by mixed variables, presenting additional challenges to the algorithm. The set of discrete values can be obtained through the following formula:

$$S_j = x_j^{\min} + (x_j^{\max} - x_j^{\min}) \cdot Q \tag{11}$$

where Q is the set of discrete values, either uniform or non-uniform. if type is uniform, $Q_u = \{0.05 * i, i = 1, ..., 20\}$, otherwise, the non-uniform set is calculated as follows:

$$Q_{nu} = \{\frac{1}{1 + e^{(5-0.5*i)}}, i = 1, ..., 20\}. \tag{12}$$

When the discrete variables include positional parameters, the shape of the PF becomes discretized. Uniformly discretized values result in a PF that is also uniformly discretized, while the opposite leads to non-uniformly discretized PF, as illustrated in Fig.4. In summary, basic benchmarks include ZDT (2 objectives) and DTLZ (3 objectives). The number of decision variables is 10, and the types of discrete values are uniform and non-uniform. The proportion of discrete variables is 0.2, 0.5, and 0.8. The maximum allowed number of function evaluations is 300.

In order to assess the performance of our proposed algorithm, we compare it with recent algorithms designed for solving EMVMOPs, including MO-ASMOCH [20], MixMOBO

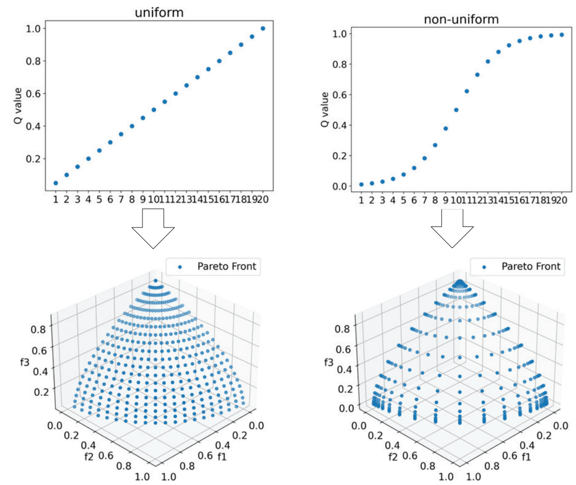

Fig. 4. An example of the influence of different types of discrete value sets on the PF of the DTLZ2 problem.

[15], and MVMOO [3], as well as the PB-NSGAIII [16] algorithm designed for continuous problems. To ensure fairness in the comparison, the maximum number of function evaluations for all algorithms is set to 300, and the initial population size is fixed at 100. The number of reference vectors is set to 100, 91, respectively, when the number of objectives is 2 and 3. For our algorithm, the parameter w_{max} is set to 50, and the default number K of re-evaluated solutions in local search is 4. The GPR model employs the implementation here [27] for handling mixed variables. Specifically, as the benchmarks do not involve constraints, the constraint-handling part is removed from MO-ASMOCH, and the resampling percentage pt is set to 0.05. In MixMOBO, the parameter Q is set to 5. To enable MVMOO to handle the benchmark problems , we replaced the original Halton sequence sampling with a genetic algorithm. In PB-NSGAIII, the relaxation method is employed to handle discrete variables. The inverted generational distance (IGD) metric [28] is used as a performance indicator for assessing algorithm performance, and the algorithm is independently run 20 times in the experiments.

B. Experiment Results and Analysis

To investigate the effects of the two core components in our algorithm, we compare ICE-MOEAmv with the other two variants, which are described as follows:

- **ICE-MOEAmv-global** utilizes only infill criterion ensemble-based sampling.
- **ICE-MOEAmv-local** employs only local search with hybrid operators (K=5).

The experiments are conducted on DTLZ2 and DTLZ3 (the discrete ratio is 0.2 and discrete values are uniform), where the former emphasizes algorithm diversity, and the latter tests algorithm convergence. In this experiment, the convergence curves of IGD values are shown in Fig.5. It can be observed that ICE-MOEAmv-global achieves faster and better convergence in terms of IGD values on the DTLZ2 problem (easy

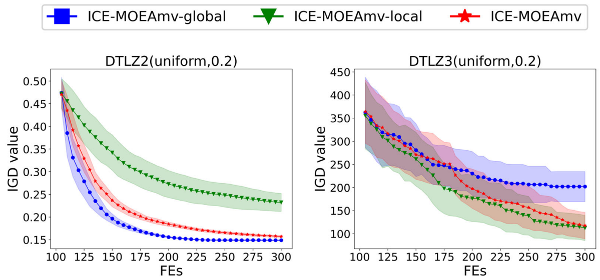

Fig. 5. The IGD convergence curve of ICE-MOEAmv and its variants on DTLZ2(uniform, 0.2) and DTLZ3(uniform, 0.2).

to converge) compared to ICE-MOEAmv-local. However, on the DTLZ3 problem (hard to converge), the situation is exactly the opposite. It can be concluded that infill criterion ensemble-based sampling plays a crucial role in enhancing algorithm diversity, while local search with hybrid operators contributes to improving algorithm convergence. As shown in Fig.6, ICE-MOEAmv-global obtains the non-dominated front with the best diversity on DTLZ2(uniform,0.2). ICE-MOEAmv, which

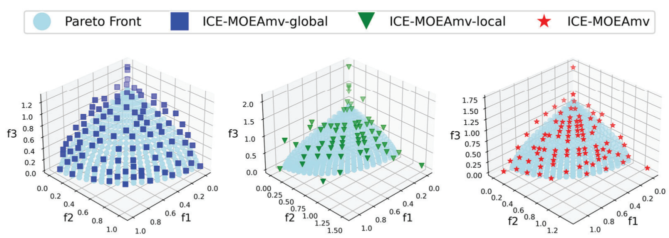

Fig. 6. The non-dominated front obtained by ICE-MOEAmv and its variants on DTLZ2(uniform,0.2) in the run associated with median IGD value.

possesses both core components, not only maintains diversity on the DTLZ2 problem but also ensures strong convergence on the DTLZ3 problem. The algorithm has a hyperparameter K during local search, which represents the number of solutions to be re-evaluated. We tested the influence of this parameter under different values $\{1, 2, 3, 4(default)\}$ on the algorithm on DTLZ2 and DTLZ3 problems. The experimental results are displayed in Fig.7. From the results, it can be seen that

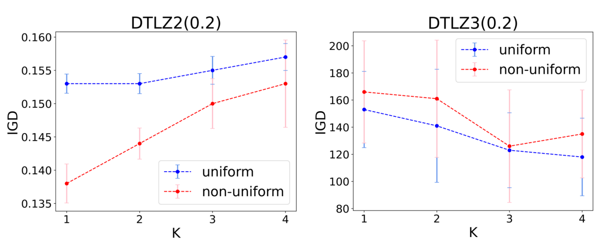

Fig. 7. The IGD results of ICE-MOEAmv on DTLZ2 and DTLZ3 for different values of the parameter $K \in \{1, 2, 3, 4(default)\}$.

as K increases, the algorithm performs worse on DTLZ2 but better on DTLZ3, which implies that increasing K would decrease algorithm diversity and enhance convergence. Actually, increasing K is equivalent to increasing the sampling ratio for local search, which strengthens local convergence. Thus, the observed results are reasonable.

To further assess the effectiveness of our proposed algorithm, we conducted experiments comparing it with four other algorithms on modified ZDT and DTLZ problems over 20 independent runs. The mean and standard deviation values of IGD with 20 runs are shown in Tables I and II, where the smallest mean value is highlighted in bold and with gray fill. In addition, the Wilcoxon rank sum test is used to test the significant, with a significance level set at 0.05. The symbols '+', '≈', '-', indicate that ICE-MOEAmv is statistically significantly superior to, almost equivalent to, and inferior to the compared algorithm, respectively. From the results in Table I, it can be observed that our algorithm outperforms the other compared algorithms on most of the modified ZDT problems. Different algorithms exhibit differences in results on problems of the same type with different discrete value types. For instance, MixMOBO performs well on ZDT6 with uniform values but is not as effective as our proposed algorithm on the non-uniform type. The MVMOO demonstrates outstanding performance on the more complex ZDT3 and ZDT4 problems, indicating that considering the uncertainty of the model contributes to solving relatively complex problems. The experimental results of the algorithms on modified DTLZ problems are presented in Table II. Our algorithm also outperforms the other compared algorithms on most modified DTLZ problems. Note that the three algorithms (MOASM-OCH, MixMOBO, and MVMOO) specifically designed for EMVMOPs do not perform well on the 3-objective problems. The PB-NSGAIII with relaxation method demonstrates advantages on the DTLZ2 and DTLZ5 problems that emphasize diversity. Our proposed algorithm, with the default setting of K as 4, prioritizes convergence more, leading to a slight disadvantage on these easy-to-converge problems. If conditions permit, we can improve our algorithm's performance on such problems by adjusting K, as discussed earlier. In summary, our proposed algorithm is competitive compared to the other algorithms on the benchmark problems.

V. Conclusion and Future Work

In this study, we propose an algorithm named ICE-MOEAmv for solving EMVMOPs. We design an infill criterion ensemble method for global reference vector-guided search, which considers the convergence to non-dominated front, model uncertainty, and diversity comprehensively through adaptive weighting of EIM and DI. Furthermore, we propose local search using hybrid operators for handling mixed variables and enhance local convergence through $I_{\epsilon+}$-based environmental selection. Our proposed algorithm is compared with other algorithms on the modified ZDT and DTLZ problems. The experimental results show that our algorithm outperforms other algorithms on most benchmark problems. The infill criterion ensemble-based sampling plays a crucial role in enhancing algorithm diversity, while local search with hybrid operators contributes to improving algorithm convergence.

TABLE I

STATISTICAL IGD RESULTS (THE MEAN AND STANDARD DEVIATION) OF ICE-MOEAMV AND FOUR COMPARED ALGORITHMS ON MODIFIED ZDT PROBLEMS BY WILCOXON RANK SUM TEST (THE SIGNIFICANCE LEVEL IS 0.05). THE BEST RESULTS ARE HIGHLIGHTED AND WITH GRAY FILL.

	ICE-MOEAmv	PBNSGAIII	MOASMO	MixMOBO	MVMOO
ZDT1(uniform,0.2)	**4.07E-10(8.02E-10)**	2.97E-03(1.58E-03)+	2.20E-02(4.55E-03)+	2.98E-02(2.97E-02)+	1.93E-01(3.82E-02)+
ZDT1(uniform,0.5)	**8.78E-11(1.67E-10)**	2.45E-03(1.90E-03)+	1.75E-02(5.00E-03)+	6.08E-02(9.87E-02)+	2.03E-01(5.44E-02)+
ZDT1(uniform,0.8)	**1.13E-11(2.65E-11)**	9.78E-04(7.70E-04)+	1.16E-02(4.59E-03)+	6.13E-02(8.70E-02)+	1.99E-01(5.04E-02)+
ZDT2(uniform,0.2)	**3.46E-10(3.57E-10)**	5.00E-03(1.53E-02)+	3.19E-02(4.41E-03)+	1.63E-02(1.47E-02)+	3.67E-01(1.43E-01)+
ZDT2(uniform,0.5)	**9.41E-11(2.08E-10)**	4.90E-03(1.92E-02)+	1.60E-02(3.99E-03)+	1.37E-02(1.20E-02)+	3.48E-01(1.49E-01)+
ZDT2(uniform,0.8)	**1.69E-11(3.00E-11)**	2.62E-02(6.87E-02)+	5.89E-03(3.26E-03)+	8.62E-03(5.68E-03)+	3.89E-01(1.40E-01)+
ZDT3(uniform,0.2)	**1.07E-02(2.15E-02)**	3.57E-02(3.96E-02)+	9.88E-02(4.61E-02)+	8.72E-02(1.18E-01)+	4.19E-02(3.02E-02)+
ZDT3(uniform,0.5)	**1.67E-02(2.49E-02)**	3.35E-02(4.79E-02)+	1.00E-01(4.61E-02)+	1.37E-01(1.48E-01)+	3.95E-02(2.31E-02)+
ZDT3(uniform,0.8)	**5.05E-13(2.05E-12)**	2.25E-02(2.04E-02)+	9.82E-02(5.83E-02)+	9.99E-02(1.02E-01)+	4.00E-02(2.36E-02)+
ZDT4(uniform,0.2)	**1.92E+01(7.32E+00)**	4.05E+01(1.06E+01)+	4.43E+01(1.05E+01)+	3.13E+01(1.17E+01)+	3.53E+01(7.27E+00)+
ZDT4(uniform,0.5)	**1.39E+01(4.61E+00)**	2.04E+01(9.30E+00)≈	2.15E+01(8.34E+00)+	1.43E+01(8.72E+00)≈	1.70E+01(3.87E+00)≈
ZDT4(uniform,0.8)	5.03E+00(2.50E+00)	**3.25E+00(2.54E+00)**≈	5.70E+00(2.66E+00)≈	5.78E+00(2.70E+00)≈	4.49E+00(1.49E+00)≈
ZDT6(uniform,0.2)	**5.39E-03(2.53E-03)**	8.87E-02(8.33E-02)+	4.63E-01(1.72E-01)+	2.31E-01(1.77E-01)+	3.46E-01(1.98E-01)+
ZDT6(uniform,0.5)	6.03E-02(1.02E-01)	7.39E-02(7.63E-02)≈	1.87E-01(1.82E-01)+	**4.69E-02(7.40E-02)-**	3.52E-01(1.72E-01)+
ZDT6(uniform,0.8)	9.92E-02(1.70E-01)	1.59E-01(1.52E-01)+	1.72E-01(1.67E-01)+	**4.27E-02(7.46E-02)-**	3.22E-01(1.79E-01)+
ZDT1(non-uniform,0.2)	**1.69E-05(5.31E-05)**	1.88E-02(7.56E-03)+	4.45E-02(1.36E-02)+	3.65E-02(1.81E-02)+	1.19E-01(2.86E-02)+
ZDT1(non-uniform,0.5)	**1.73E-04(1.89E-04)**	1.83E-02(8.81E-03)+	5.14E-02(1.86E-02)+	2.41E-02(1.43E-02)+	4.00E-02(9.34E-03)+
ZDT1(non-uniform,0.8)	**3.96E-03(2.90E-03)**	1.87E-02(7.79E-03)+	4.75E-02(2.37E-02)+	1.07E-01(2.03E-01)+	2.06E-02(7.61E-03)+
ZDT2(non-uniform,0.2)	**1.27E-08(1.90E-08)**	6.36E-02(1.57E-01)+	3.26E-02(5.68E-03)+	3.58E-02(2.32E-02)+	3.44E-01(2.18E-01)+
ZDT2(non-uniform,0.5)	**7.13E-04(1.10E-03)**	9.89E-02(2.01E-01)+	2.86E-02(8.70E-03)+	4.72E-01(2.73E-01)+	6.62E-02(2.43E-02)+
ZDT2(non-uniform,0.8)	**1.03E-02(1.32E-02)**	1.76E-01(2.65E-01)+	2.41E-02(5.37E-03)+	8.18E-01(2.18E-01)+	2.58E-02(7.02E-03)+
ZDT3(non-uniform,0.2)	**4.26E-03(1.33E-02)**	8.95E-02(1.21E-01)+	7.06E-02(2.18E-02)+	2.82E-01(2.02E-01)+	1.66E-02(1.06E-02)+
ZDT3(non-uniform,0.5)	2.43E-02(2.62E-02)	8.95E-02(9.18E-02)+	7.30E-02(2.33E-02)+	1.11E+00(3.69E-01)+	**1.08E-02(8.56E-03)**≈
ZDT3(non-uniform,0.8)	3.18E-02(2.35E-02)	1.04E-01(1.10E-01)+	7.66E-02(2.84E-02)+	1.09E+00(4.55E-01)+	**1.18E-02(7.16E-03)-**
ZDT4(non-uniform,0.2)	**2.59E+01(9.51E+00)**	4.12E+01(1.50E+01)+	5.30E+01(1.13E+01)+	4.57E+01(1.92E+01)+	4.18E+01(8.70E+00)+
ZDT4(non-uniform,0.5)	5.16E+01(1.25E+01)	5.21E+01(1.62E+01)≈	5.70E+01(1.46E+01)≈	6.16E+01(2.17E+01)≈	**4.62E+01(1.56E+01)**≈
ZDT4(non-uniform,0.8)	7.39E+01(1.87E+01)	5.82E+01(1.91E+01)-	6.18E+01(1.96E+01)≈	9.06E+01(1.97E+01)+	**4.15E+01(2.32E+01)-**
ZDT6(non-uniform,0.2)	**2.60E-03(2.29E-03)**	7.67E-02(6.24E-02)+	9.82E-01(2.58E-01)+	1.16E+00(3.91E-01)+	7.98E-01(5.38E-01)+
ZDT6(non-uniform,0.5)	**3.05E-03(1.50E-03)**	9.39E-02(1.46E-01)+	4.28E-01(2.58E-01)+	8.11E-01(4.08E-01)+	3.46E-01(3.66E-01)+
ZDT6(non-uniform,0.8)	**3.25E-03(1.73E-03)**	5.57E-02(5.15E-02)+	2.76E-01(2.73E-01)+	7.83E-01(1.78E-01)+	3.92E-01(3.42E-01)+
+/≈/-		25/4/1	27/3/0	25/3/2	24/4/2

The symbols '+', '≈', '-', indicate that ICE-MOEAmv is statistically significantly superior to, almost equivalent to, and inferior to the compared algorithm, respectively. The p-values for multiple tests were adjusted using the Benjamini-Hochberg method.

From the results of parameter sensitivity experiments, it can be observed that as K increases, the algorithm's convergence improves, but diversity decreases. Due to the default setting of K emphasizing algorithm convergence, this results in the algorithm not performing optimally on some easy-to-converge problems.

In future work, designing an adaptive strategy for selecting the parameter K is a promising direction to further enhance the algorithm's performance on all problems. Also, the algorithm is still worth investigating in problems with higher decision variable dimensions or more objectives.

REFERENCES

[1] A. Thebelt, C. Tsay, R. M. Lee, N. Sudermann-Merx, D. Walz, T. Tranter, and R. Misener, "Multi-objective constrained optimization for energy applications via tree ensembles," *Applied Energy*, vol. 306, p. 118061, 2022.

[2] C. He, Y. Zhang, D. Gong, and X. Ji, "A review of surrogate-assisted evolutionary algorithms for expensive optimization problems," *Expert Systems with Applications*, p. 119495, 2023.

[3] J. A. Manson, T. W. Chamberlain, and R. A. Bourne, "Mvmoo: Mixed variable multi-objective optimisation," *Journal of Global Optimization*, vol. 80, no. 4, pp. 865–886, 2021.

[4] Y. Liu and H. Wang, "Surrogate-assisted hybrid evolutionary algorithm with local estimation of distribution for expensive mixed-variable optimization problems," *Applied Soft Computing*, vol. 133, p. 109957, 2023.

[5] E. C. Garrido-Merchán and D. Hernández-Lobato, "Dealing with categorical and integer-valued variables in bayesian optimization with gaussian processes," *Neurocomputing*, vol. 380, pp. 20–35, 2020.

[6] A. Zhou, J. Sun, and Q. Zhang, "An estimation of distribution algorithm with cheap and expensive local search methods," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 807–822, 2015.

[7] H. Peng, Y. Han, C. Deng, J. Wang, and Z. Wu, "Multi-strategy co-evolutionary differential evolution for mixed-variable optimization," *Knowledge-Based Systems*, vol. 229, p. 107366, 2021.

[8] F. Wang, H. Zhang, and A. Zhou, "A particle swarm optimization algorithm for mixed-variable optimization problems," *Swarm and Evolutionary Computation*, vol. 60, p. 100808, 2021.

[9] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[10] Z. Song, H. Wang, C. He, and Y. Jin, "A kriging-assisted two-archive evolutionary algorithm for expensive many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 6, pp. 1013–1027, 2021.

[11] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on evolutionary computation*, vol. 11, no. 6, pp. 712–731, 2007.

[12] T. Chugh, Y. Jin, K. Miettinen, J. Hakanen, and K. Sindhya, "A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 129–142, 2016.

[13] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *International conference on parallel problem solving from nature*. Springer, 2004, pp. 832–842.

[14] J. G. Falcón-Cardona and C. A. C. Coello, "Indicator-based multi-objective evolutionary algorithms: A comprehensive survey," *ACM Computing Surveys (CSUR)*, vol. 53, no. 2, pp. 1–35, 2020.

[15] H. M. Sheikh and P. S. Marcus, "Bayesian optimization for mixed-variable, multi-objective problems," *Structural and Multidisciplinary Optimization*, vol. 65, no. 11, p. 331, 2022.

[16] Z. Song, H. Wang, and H. Xu, "A framework for expensive many-

TABLE II

STATISTICAL IGD RESULTS (THE MEAN AND STANDARD DEVIATION) OF ICE-MOEAMV AND FOUR COMPARED ALGORITHMS ON MODIFIED DTLZ PROBLEMS BY WILCOXON RANK SUM TEST (THE SIGNIFICANCE LEVEL IS 0.05). THE BEST RESULTS ARE HIGHLIGHTED AND WITH GRAY FILL.

	ICE-MOEAmv	PBNSGAIII	MOASMO	MixMOBO	MVMOO
DTLZ1(uniform,0.2)	**5.72E+01(1.69E+01)**	5.76E+01(2.14E+01)≈	1.06E+02(3.13E+01)+	8.93E+01(2.10E+01)+	7.17E+01(9.97E+00)+
DTLZ1(uniform,0.5)	3.77E+01(1.01E+01)	**3.32E+01(1.04E+01)≈**	8.61E+01(2.29E+01)+	7.73E+01(2.09E+01)+	5.47E+01(1.06E+01)+
DTLZ1(uniform,0.8)	**1.88E+01(5.20E+00)**	2.15E+01(5.75E+00)≈	4.85E+01(1.99E+01)+	6.11E+01(1.88E+01)+	5.83E+01(1.55E+01)+
DTLZ2(uniform,0.2)	**1.57E-01(2.02E-03)**	1.57E-01(3.13E-03)≈	3.02E-01(1.63E-02)+	3.28E-01(2.46E-02)+	2.14E-01(9.63E-03)+
DTLZ2(uniform,0.5)	1.61E-01(1.91E-03)	**1.58E-01(4.25E-03)-**	3.18E-01(1.70E-02)+	3.33E-01(1.92E-02)+	2.12E-01(8.43E-03)+
DTLZ2(uniform,0.8)	1.64E-01(2.50E-03)	**1.55E-01(2.24E-03)-**	3.07E-01(2.23E-02)+	3.37E-01(1.82E-02)+	2.06E-01(6.90E-03)+
DTLZ3(uniform,0.2)	**1.18E+02(2.86E+01)**	1.21E+02(3.65E+01)≈	2.95E+02(5.42E+01)+	2.30E+02(2.88E+01)+	1.78E+02(2.83E+01)+
DTLZ3(uniform,0.5)	**7.34E+01(2.18E+01)**	9.56E+01(3.73E+01)≈	2.06E+02(5.10E+01)+	2.11E+02(5.02E+01)+	1.64E+02(5.28E+01)+
DTLZ3(uniform,0.8)	**4.16E+01(1.14E+01)**	6.18E+01(1.80E+01)+	1.32E+02(6.70E+01)+	1.63E+02(6.56E+01)+	1.53E+02(6.16E+01)+
DTLZ4(uniform,0.2)	**4.95E-01(7.70E-03)**	5.27E-01(2.96E-02)+	5.77E-01(2.21E-02)+	7.26E-01(5.86E-02)+	6.93E-01(5.87E-02)+
DTLZ4(uniform,0.5)	**5.01E-01(1.21E-02)**	5.18E-01(1.96E-02)+	5.74E-01(2.60E-02)+	7.32E-01(4.91E-02)+	7.09E-01(6.44E-02)+
DTLZ4(uniform,0.8)	**5.04E-01(1.96E-02)**	5.17E-01(3.17E-02)≈	5.74E-01(3.17E-02)+	7.29E-01(5.03E-02)+	7.04E-01(4.20E-02)+
DTLZ5(uniform,0.2)	**1.35E-01(3.36E-03)**	1.35E-01(7.14E-04)+	1.88E-01(9.82E-03)+	2.16E-01(2.69E-02)+	1.78E-01(7.97E-03)+
DTLZ5(uniform,0.5)	**1.34E-01(8.32E-04)**	1.35E-01(1.52E-03)+	1.91E-01(1.32E-02)+	2.28E-01(2.79E-02)+	1.75E-01(8.49E-03)+
DTLZ5(uniform,0.8)	1.36E-01(3.18E-03)	**1.35E-01(1.21E-03)≈**	1.98E-01(2.05E-02)+	2.22E-01(2.29E-02)+	1.76E-01(8.26E-03)+
DTLZ6(uniform,0.2)	**7.91E-01(3.01E-01)**	2.65E+00(5.26E-01)+	4.00E+00(5.05E-01)+	2.52E+00(6.90E-01)+	1.06E+00(2.82E-01)+
DTLZ6(uniform,0.5)	**2.48E+00(1.26E-01)**	3.53E+00(3.51E-01)+	4.51E+00(5.31E-01)+	3.72E+00(4.10E-01)+	2.58E+00(9.70E-02)+
DTLZ6(uniform,0.8)	**4.60E+00(1.04E-02)**	4.93E+00(1.90E-01)+	5.38E+00(1.65E-01)+	5.26E+00(1.68E-01)+	4.60E+00(1.79E-02)≈
DTLZ7(uniform,0.2)	**8.63E-02(9.83E-03)**	2.04E-01(1.52E-01)≈	4.48E-01(6.12E-02)+	2.34E-01(8.24E-02)+	4.06E-01(8.83E-02)+
DTLZ7(uniform,0.5)	**2.01E-01(1.28E-03)**	3.14E-01(2.39E-01)+	5.19E-01(8.60E-02)+	2.96E-01(6.14E-02)+	5.39E-01(8.98E-02)+
DTLZ7(uniform,0.8)	**4.25E-01(1.01E-03)**	5.34E-01(1.74E-01)+	6.69E-01(1.08E-01)+	4.92E-01(3.74E-02)+	7.27E-01(1.17E-01)+
DTLZ1(non-uniform,0.2)	**6.33E+01(1.73E+01)**	6.78E+01(2.33E+01)≈	1.12E+02(3.16E+01)+	9.73E+01(2.19E+01)+	7.24E+01(1.12E+01)≈
DTLZ1(non-uniform,0.5)	**7.52E+01(1.99E+01)**	7.83E+01(1.75E+01)≈	1.21E+02(2.25E+01)+	9.74E+01(1.48E+01)+	8.85E+01(9.10E+00)+
DTLZ1(non-uniform,0.8)	1.00E+02(1.51E+01)	**8.07E+01(1.68E+01)-**	1.47E+02(2.28E+01)+	1.03E+02(1.43E+01)≈	9.38E+01(7.73E+00)≈
DTLZ2(non-uniform,0.2)	1.53E-01(6.56E-03)	**1.50E-01(9.94E-03)-**	3.14E-01(2.93E-02)+	2.95E-01(1.99E-02)+	2.23E-01(1.02E-02)+
DTLZ2(non-uniform,0.5)	2.16E-01(2.35E-02)	**1.75E-01(1.40E-02)-**	4.24E-01(3.41E-02)+	4.18E-01(4.17E-02)+	3.00E-01(2.49E-02)+
DTLZ2(non-uniform,0.8)	2.93E-01(3.62E-02)	**2.21E-01(3.04E-02)-**	5.54E-01(5.07E-02)+	6.18E-01(6.36E-02)+	4.40E-01(5.49E-02)+
DTLZ3(non-uniform,0.2)	**1.35E+02(3.24E+01)**	1.98E+02(6.36E+01)+	2.89E+02(7.46E+01)+	3.15E+02(4.82E+01)+	1.76E+02(2.72E+01)+
DTLZ3(non-uniform,0.5)	**1.57E+02(4.93E+01)**	2.15E+02(4.63E+01)+	3.23E+02(4.87E+01)+	3.69E+02(5.34E+01)+	2.05E+02(2.16E+01)+
DTLZ3(non-uniform,0.8)	**2.30E+02(5.41E+01)**	2.56E+02(4.09E+01)≈	3.72E+02(4.04E+01)+	3.70E+02(4.66E+01)+	2.39E+02(2.37E+01)≈
DTLZ4(non-uniform,0.2)	**4.52E-01(3.64E-03)**	4.67E-01(7.16E-02)≈	4.98E-01(2.07E-02)+	4.82E-01(4.92E-03)+	5.24E-01(2.15E-02)+
DTLZ4(non-uniform,0.5)	**4.73E-01(2.31E-02)**	4.84E-01(7.51E-02)≈	5.23E-01(1.78E-02)+	5.08E-01(1.07E-02)+	5.85E-01(3.90E-02)+
DTLZ4(non-uniform,0.8)	**4.96E-01(2.31E-02)**	5.96E-01(1.34E-01)≈	5.72E-01(5.09E-02)+	5.63E-01(4.28E-02)+	6.87E-01(6.84E-02)+
DTLZ5(non-uniform,0.2)	**1.23E-01(2.47E-03)**	1.24E-01(5.15E-03)≈	2.35E-01(2.57E-02)+	3.70E-01(3.81E-02)+	1.69E-01(1.31E-02)+
DTLZ5(non-uniform,0.5)	1.56E-01(1.26E-02)	**1.49E-01(1.72E-02)≈**	3.51E-01(3.67E-02)+	5.28E-01(7.18E-02)+	2.49E-01(4.85E-02)+
DTLZ5(non-uniform,0.8)	2.06E-01(2.99E-02)	**1.63E-01(1.75E-02)-**	5.08E-01(7.22E-02)+	6.59E-01(6.65E-02)+	3.72E-01(4.56E-02)+
DTLZ6(non-uniform,0.2)	**5.88E-01(2.98E-01)**	3.09E+00(6.06E-01)+	5.66E+00(3.17E-01)+	4.88E+00(2.85E-01)+	1.22E+00(3.29E-01)+
DTLZ6(non-uniform,0.5)	**2.17E+00(2.46E-01)**	3.62E+00(3.98E-01)+	5.53E+00(3.68E-01)+	5.04E+00(3.52E-01)+	2.31E+00(1.87E-01)+
DTLZ6(non-uniform,0.8)	**3.95E+00(2.02E-02)**	4.32E+00(2.66E-01)+	5.97E+00(2.88E-01)+	5.35E+00(3.65E-01)+	3.99E+00(4.06E-02)+
DTLZ7(non-uniform,0.2)	**8.49E-02(7.50E-03)**	2.75E-01(1.77E-01)+	5.02E-01(6.65E-02)+	1.23E+00(8.74E-01)+	2.94E-01(4.43E-02)+
DTLZ7(non-uniform,0.5)	**1.31E-01(1.42E-02)**	4.42E-01(2.05E-01)+	4.84E-01(8.11E-02)+	3.03E+00(2.13E+00)+	2.26E-01(5.91E-02)+
DTLZ7(non-uniform,0.8)	**2.28E-01(5.42E-02)**	4.58E-01(2.08E-01)+	5.43E-01(1.16E-01)+	3.30E+00(2.00E+00)+	2.54E-01(3.55E-02)+
+/≈/-		18/17/7	42/0/0	41/1/0	38/4/0

The symbols '+', '≈', '-', indicate that ICE-MOEAmv is statistically significantly superior to, almost equivalent to, and inferior to the compared algorithm, respectively. The p-values for multiple tests were adjusted using the Benjamini-Hochberg method.

objective optimization with pareto-based bi-indicator infill sampling criterion," *Memetic Computing*, pp. 1–13, 2022.

[17] S. Qin, C. Sun, Q. Liu, and Y. Jin, "A performance indicator based infill criterion for expensive multi-/many-objective optimization," *IEEE transactions on evolutionary computation*, 2023.

[18] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints," *IEEE transactions on evolutionary computation*, vol. 18, no. 4, pp. 577–601, 2013.

[19] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.

[20] R. Sun, Q. Duan, and X. Mao, "A multi-objective adaptive surrogate modelling-based optimization algorithm for constrained hybrid problems," *Environmental Modelling & Software*, vol. 148, p. 105272, 2022.

[21] A. Habib, H. K. Singh, T. Chugh, T. Ray, and K. Miettinen, "A multiple surrogate assisted decomposition-based evolutionary algorithm for expensive multi/many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 6, pp. 1000–1014, 2019.

[22] H. Wang, L. Jiao, and X. Yao, "Two_arch2: An improved two-archive algorithm for many-objective optimization," *IEEE transactions on evolutionary computation*, vol. 19, no. 4, pp. 524–541, 2014.

[23] Y. Jin, H. Wang, T. Chugh, D. Guo, and K. Miettinen, "Data-driven evolutionary optimization: An overview and case studies," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 442–458, 2018.

[24] D. Zhan, Y. Cheng, and J. Liu, "Expected improvement matrix-based infill criteria for expensive multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 6, pp. 956–975, 2017.

[25] U. Škvorc, T. Eftimov, and P. Korošec, "The effect of sampling methods on the invariance to function transformations when using exploratory landscape analysis," in *2021 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2021, pp. 1139–1146.

[26] W. Wang, K. Li, H. Jalil, and H. Wang, "An improved estimation of distribution algorithm for multi-objective optimization problems with mixed-variable," *Neural Computing and Applications*, vol. 34, no. 22, pp. 19 703–19 721, 2022.

[27] P. Saves, R. Lafage, N. Bartoli, Y. Diouane, J. Bussemaker, T. Lefebvre, J. T. Hwang, J. Morlier, and J. R. Martins, "Smt 2.0: A surrogate modeling toolbox with a focus on hierarchical and mixed variables gaussian processes," *arXiv preprint arXiv:2305.13998*, 2023.

[28] P. A. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE transactions on evolutionary computation*, vol. 7, no. 2, pp. 174–188, 2003.