Proceedings of the 2025 SICE Festival with Annual Conference
September 9-12, 2025, Chiang Mai, Thailand

Improved Infill Criteria for Evolutionary Surrogate-Assisted Optimization of Highly Expensive Multiobjective Problems

Ryo Fukami [1†] and Masaya Nakata [1]

[1]Faculty of Engineering, Yokohama National University, Kanagawa, Japan
(Tel: +81-45-339-4139; E-mail: fukami-ryo-mn@ynu.jp, nakata-masaya-tb@ynu.ac.jp)

Abstract: Many real-world applications involve highly expensive multiobjective optimization problems (MOPs), where the number of allowable function evaluations is severely limited to only a few dozen. This paper proposes a surrogate-assisted multiobjective evolutionary algorithm (SAMOEA) specifically designed for such challenging scenarios. We introduce a novel infill criterion that selects most effective candidate solutions by rigorously evaluating their contributions to both convergence and diversity, and integrate it into an SAMOEA framework. Experimental results demonstrate that the proposed algorithm significantly outperforms existing SAMOEAs under a strict budget of 50 function evaluations. Moreover, our analysis reveals that the proposed infill criterion improves the sampling efficiency, thereby contributing to the improved performance.

Keywords: Surrogate-assisted evolutionary computation, highly expensive multiobjective optimization, infill criteria.

1. INTRODUCTION

Many real-world optimization problems involve multiple expensive-to-evaluate objectives, with a limited number of allowable function evaluations (FEs). These problems are referred to as expensive multiobjective optimization problems (expensive MOPs), typically assuming a few hundred FEs [1]. However, in practice, even a few hundred FEs is still a large budget for certain engineering problems, such as manufacturing and drug design [2–4]. For example, a car crash simulation takes more than 20 hours to evaluate one solution [5]. In such extreme cases, namely *highly expensive* MOPs, the number of FEs may be restricted to several dozen [6].

Surrogate-assisted multiobjective evolutionary algorithms (SAMOEAs) [7] are powerful approaches for solving expensive MOPs. Typically, surrogate models are constructed as regression models of the objective functions or other performance metrics, where evaluated solutions are used as training data. Commonly used surrogate models include Gaussian processes (GPs) [8], radial basis functions (RBFs) [9], and artificial neural networks (ANNs) [10]. Similar to Bayesian optimization, SAMOEAs utilize surrogate models to estimate the quality of solutions, and then select promising candidates for evaluation with expensive objective functions, guided by infill criteria. Infill criteria, also known as acquisition functions, are conditions or functions that define which solutions should be selected for actual evaluation. In this way, SAMOEAs accelerate the evolutionary search under a limited budget of FEs.

However, most SAMOEAs assume a budget of several hundred FEs [11], making them somewhat impractical for solving highly expensive MOPs. As the number of FEs is severely limited to several dozen, typical SAMOEAs suffer from a significant degradation in sampling efficiency due to the following difficulties. First, constructing effective surrogate models is challenging, even with many pre-evaluated solutions (i.e., initial training data). This is because surrogate models usually need to be adapted online to reflect the current search dynamics; however, this is hindered by the limited number of FEs, that is, the scarcity of online-sampled training data. Second, many SAMOEAs employ infill criteria designed to achieve a good balance between exploration and exploitation [12–14], but this reduces the convergence speed of algorithms in highly expensive MOPs.

Accordingly, we focus on improving infill criteria tailored to highly expensive multiobjective problems (MOPs), rather than improving the quality of surrogate models due to their complexity. Specifically, this paper proposes an extension of an RBF-based SAMOEA, SFA/DE [15]. We develop and integrate a rigorous infill criterion into the SFA/DE framework. Our infill criterion is designed to identify more promising solutions by rigorously evaluating their contributions to both the diversity and convergence of Pareto solutions, thereby increasing the bias of exploitation. Furthermore, our proposed algorithm, called SFA/DE-IC, skips evaluating solutions with the expensive objective functions if no candidate solutions satisfy the infill criterion, to save the expensive FEs. The contributions of this paper are summarized below.

1. This paper demonstrates an effective design of infill criteria for highly expensive MOPs. We show that SFA/DE becomes more practical in this complex problem class by selecting more promising solutions using our infill criterion.
2. To the best of our knowledge, this paper presents the first comparative study investigating the effectiveness of state-of-the-art SAMOEAs under extremely limited budgets of FEs. We compare SFA/DE-IC with seven state-of-the-art SAMOEAs.

The rest of this paper is organized as follows. Section 2 provides background information on RBF and SFA/DE. Section 3 introduces our infill criterion and the SFA/DE-IC framework. Section 4 presents experimental results to

† Ryo Fukami is the presenter of this paper.

evaluate the effectiveness of SFA/DE-IC in highly expensive MOPs. Section 5 presents ablation studies to assess the impact of our infill criterion. Finally, Section 6 summarizes the contributions of this work.

2. BACKGROUND

2.1. Radial basis function (RBF)

An RBF is a popular regression model. Given N training samples $\{(\boldsymbol{x}^i, f(\boldsymbol{x}^i))\}_{i=1}^N$, where $\boldsymbol{x}^i \in \mathbb{R}^D$, $f(\boldsymbol{x}^i) \in \mathbb{R}$ denotes the target values of $\boldsymbol{x}^i$, an RBF model approximates the function f, denoted by $\hat{f}$, as follows:

$$\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{N} w_i \Phi(\|\boldsymbol{x} - \boldsymbol{x}^i\|), \tag{1}$$

where w_i is the weight coefficient, and $\|\boldsymbol{x} - \boldsymbol{x}^i\|$ is the Euclidean distance between a given point $\boldsymbol{x}$ and $\boldsymbol{x}^i$. $\Phi(\cdot)$ expresses a kernel function. In this study, we use a Gaussian function, expressed as $\Phi(r) = \exp\left(-r^2/\sigma^2\right)$, where σ^2 is defined as $d_{max}(ND)^{-1/D}$ [15] with the maximum distance between training samples d_{max}. A weight vector, $\boldsymbol{w} = [w_1, w_2, \ldots, w_N]^T$ is computed as $\boldsymbol{w} = \boldsymbol{\Phi}^{-1}\boldsymbol{F}$, where $\boldsymbol{F} = [f(x^1), f(x^2), \ldots, f(x^n)]^T$ and $\boldsymbol{\Phi}$ is a kernel matrix, given by

$$\boldsymbol{\Phi} = \begin{bmatrix} \Phi(\boldsymbol{x}^1, \boldsymbol{x}^1) & \cdots & \Phi(\boldsymbol{x}^1, \boldsymbol{x}^N) \\ \vdots & \ddots & \vdots \\ \Phi(\boldsymbol{x}^N, \boldsymbol{x}^1) & \cdots & \Phi(\boldsymbol{x}^N, \boldsymbol{x}^N) \end{bmatrix}. \tag{2}$$

2.2. SFA/DE

SFA/DE [15] is a state-of-the-art SAMOEA based on a decomposition framework of MOEA/D [16]. We first explain this decomposition framework as background information. Subsequently, the detailed SFA/DE framework is explained.

2.2.1. Decomposition framework

To begin with, this paper considers MOPs in the minimization form, given by

$$\begin{aligned} &\text{minimize} && \boldsymbol{F}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_M(\boldsymbol{x})], \\ &\text{subject to} && \boldsymbol{x} \in \mathcal{S}, \end{aligned} \tag{3}$$

where $M(\geq 2)$ is the number of objective functions, $\mathcal{S} \subseteq \mathbb{R}^D$ is a feasible space, and D is the number of decision variables. A main goal of MOPs is to approximate the true Pareto front, which involves a set of non-dominated solutions (called Pareto optimal solutions). A dominance relationship is defined as follows: a solution $\boldsymbol{u}$ *dominates* a solution $\boldsymbol{v}$, denoted as $\boldsymbol{u} \prec \boldsymbol{v}$, if $f_i(\boldsymbol{u}) \leq f_i(\boldsymbol{v}),\ \forall\, i \in \{1, \ldots, M\}$ and $f_j(\boldsymbol{u}) < f_j(\boldsymbol{v}),\ \exists\, j \in \{1, \ldots, M\}$.

The decomposition approach is a widely used strategy for solving MOPs. It decomposes an MOP into a set of N single-objective subproblems using scalarization functions, where N typically corresponds to the population size of an optimizer. Each subproblem is defined by its corresponding scalarization function $g : \mathbb{R}^{\mathbb{D}} \to \mathbb{R}$, and the goal is to minimize this single-objective function. SFA/DE employs the Tchebycheff scalarization function, defined as

$$g(\boldsymbol{x} \mid \boldsymbol{\lambda}^i, \boldsymbol{z}) = \max_{1 \leq j \leq M} \left\{\lambda_j^i |f_j(\boldsymbol{x}) - z_j|\right\}, \tag{4}$$

for the i-th subproblem, where $\boldsymbol{z} = \{z_1, \ldots, z_M\}$ is a reference point, and $\boldsymbol{\lambda}^i = \{\lambda_1^i, \ldots, \lambda_M^i\}$ is a weight vector indicating a search direction.

2.2.2. Algorithm

While the original SFA/DE framework was developed under the assumption that no pre-evaluated solutions are available in advance, this paper assumes the availability of a large number of such solutions, as this situation frequently arises in real-world applications. These pre-evaluated solutions can serve both as training samples for building surrogate models and as initial solutions for the population $\mathcal{P}$. Algorithm 1 shows the pseudocode of SFA/DE, which has been modified for our assumption; unlike its original description, a set of pre-evaluated solutions, denoted by $\mathcal{T}$, with $|\mathcal{T}| \geq \mathcal{N}$, is assumed to be externally provided as input to the algorithm.

Initially, an index set $\mathcal{B}(i)$ is defined for the i-th subproblem as $\mathcal{B}(i) = \{i_1, \ldots, i_T\}$, where $\lambda^{i_1}, \ldots, \lambda^{i_T}$ are the T weight vectors closest to λ^i. An archive set $\mathcal{A}$, which stores all evaluated solutions, is initialized as a duplicate of $\mathcal{T}$. A population $\mathcal{P}$ is initialized with N solutions selected from $\mathcal{T}$ using a defined selection scheme (see Section 4.1.2 for details). Each reference point z_j is determined as the minimum value of each objective function f_j of $\boldsymbol{x} \in \mathcal{A}$.

As a main loop, for the i-th subproblem, SFA/DE first constructs an RBF model as a regression model of the corresponding scalarization function, and then performs a Differential Evolution (DE)-based search to obtain a promising solution estimated by the RBF model. Specifically, a training dataset $\mathcal{D}^i$ is determined as a set of N solutions in $\mathcal{A}$, which have the top N minimum values of $g(\boldsymbol{x}|\boldsymbol{\lambda}^i, \boldsymbol{z})$, that is,

$$\mathcal{D}^i = \{(\boldsymbol{x}^k, g(\boldsymbol{x}^k|\boldsymbol{\lambda}^i, \boldsymbol{z})) \mid \boldsymbol{x}^k \in \mathcal{A}_i^g\}_{k=1}^N, \tag{5}$$

where $\mathcal{A}_i^g$ is a duplicate of $\mathcal{A}$, in which the solutions are sorted in ascending order by the value of $g(\boldsymbol{x}|\boldsymbol{\lambda}^i, \boldsymbol{z})$. Subsequently, SFA/DE constructs an RBF model $\hat{g}^i$ for the corresponding scalarization function g^i, which is trained for $\mathcal{D}^i$. Next, SFA/DE performs the DE algorithm to optimize $\hat{g}^i$. The DE population $\mathcal{Q}$ of size T is initialized as a set of neighbor solutions for the i-th subproblem, given by

$$\mathcal{Q} = \{\boldsymbol{x}^l \mid \boldsymbol{x}^l \in \mathcal{P},\ \forall l \in \mathcal{B}(i)\}. \tag{6}$$

Then, the DE-based search is executed for $\omega_{\max}$ generations to produce solutions in $\mathcal{Q}$ optimized for $\hat{g}^i$. In this paper, we used the current-to-best/1 mutation and binomial crossover for the DE genetic operators. After completing $\omega_{\max}$ generations, the best solution, $\boldsymbol{y}^i$, having the minimum $\hat{g}^i$ value in $\mathcal{Q}$, is selected and then evaluated with the original objective functions; and $\mathcal{A}$

Algorithm 1 SFA/DE

Input: Set of pre-evaluated solutions $\mathcal{T}$, Population size N, Maximum number of generations on DE-based search $w_{\max}$, Neighbor size T, Parameters of DE-based search F, CR
Output: Non-dominated solutions in $\mathcal{A}$
Set $\mathcal{B}(i)\ \forall i \in \{1, \dots, N\}$
Initialize an archive $\mathcal{A}$ as a duplicate of $\mathcal{T}$
Initialize a population $\mathcal{P}$ with N solutions selected from $\mathcal{T}$
Initialize $\boldsymbol{z}$ as $z_j = \min_{\boldsymbol{x}\in\mathcal{A}} f_j(\boldsymbol{x}) \forall j \in \{1, \dots, M\}$
while termination criteria are not satisfied **do**
  for $i = 1$ **to** N **do**
    Build a training dataset $\mathcal{D}^i$ with $\mathcal{A}$ by Eq. (5)
    Train an RBF model $\hat{g}^i$ with $\mathcal{D}^i$
    Initialize DE population $\mathcal{Q}$ by Eq. (6)
    for $w = 1$ **to** $w_{\max}$ **do**
      Set an offspring set $\mathcal{V}$ as an empty set
      for $j = 1$ **to** $|\mathcal{Q}|$ **do**
        Generate offspring solution $\boldsymbol{u}^j$ by applying crossover and mutation with F, CR to $\boldsymbol{x}^j \in \mathcal{Q}$
        if $\hat{g}^i(\boldsymbol{u}^j) \le \hat{g}^i(\boldsymbol{x}^j)$ **then**
          Add $\boldsymbol{u}^j$ to $\mathcal{V}$
        else
          Add $\boldsymbol{x}^j$ to $\mathcal{V}$
        end if
      end for
      Update $\mathcal{Q}$ as $\mathcal{Q} \leftarrow \mathcal{V}$
    end for
    Set $\boldsymbol{y}^i$ as $\boldsymbol{y}^i \leftarrow \arg\min_{\boldsymbol{x}\in\mathcal{Q}} \hat{g}^i(\boldsymbol{x})$
    Evaluate $\boldsymbol{y}^i$ with the original objective functions
    Replace $\boldsymbol{x}^j$ with $\boldsymbol{y}^i$ if $g(\boldsymbol{y}^i|\boldsymbol{\lambda}^j) \le g(\boldsymbol{x}^j|\boldsymbol{\lambda}^j)\ \forall j \in \mathcal{B}(i)$
    Add $\boldsymbol{y}^i$ to $\mathcal{A}$
    Update $\boldsymbol{z}$ as $z_j \leftarrow \min\{z_j, f_j(\boldsymbol{y}^i)\}\ \forall j \in \{1, \dots, M\}$
  end for
end while

and $\boldsymbol{z}$ are updated accordingly. For each neighbor solution $\boldsymbol{x}^j$, where $j \in \mathcal{B}(i)$, may be replaced with $\boldsymbol{y}^i$ if $g(\boldsymbol{y}^i|\boldsymbol{\lambda}^j, \boldsymbol{z}) \le g(\boldsymbol{x}^j|\boldsymbol{\lambda}^j, \boldsymbol{z})$. These processes are repeated for each subproblem.

3. PROPOSED ALGORITHM

The proposed algorithm, namely SFA/DE-IC, incorporates a rigorous infill criterion into the SFA/DE framework to adapt it for highly expensive MOPs. Our infill strategy consists of a two-step criterion designed to identify more promising solutions that improve both the diversity and convergence of SFA/DE under severely limited FEs.

This section begins by introducing our two-step infill criterion. Subsequently, the entire procedure of SFA/DE-IC is described.

3.1. Two-step infill criterion

As aforementioned in Section 1, typical SAMOEAs employ infill criteria designed to balance exploration and exploitation [12–14]. This approach aims to estimate a promising, unvisited region through exploration and then intensively search this region via exploitation. While this search strategy is reasonable for expensive MOPs with a few hundred FEs, it instead hinders the convergence speed of algorithms in highly expensive MOPs. Furthermore, to implement the exploitation strategy, the infill criteria of certain SAMOEAs are designed to selectively include less promising solutions. In the case of SFA/DE, although it selects and evaluates the *best* solution with the minimum $\hat{g}^i$ value in $\mathcal{Q}$, its optimality is loosely estimated, as it has not been validated whether it provides the best performance across all possible solutions in $\mathcal{A}$, or whether it becomes a Pareto solution compared to the current discovered Pareto front (see Fig.1(a)).

To address the inefficiency of SFA/DE, we propose the following two criteria to identify solutions that contribute significantly to the improvement of Pareto solutions, as described below.

- **The first criterion** is to select a candidate solution $\boldsymbol{y}^i$ that has the minimum $\hat{g}^i$ value not only among all candidate solutions in $\mathcal{Q}$ but also among all evaluated solutions stored in $\mathcal{A}$. That is, $\boldsymbol{y}^i$ should satisfy the condition

$$\nexists \boldsymbol{x} \in \mathcal{A} \cup \mathcal{Q},\ \ \hat{g}^i(\boldsymbol{x}) \le \hat{g}^i(\boldsymbol{y}^i). \tag{7}$$

This criterion aims to improve the convergence speed of algorithms by selecting a well-converged solution. Fig. 1(b) shows a counterexample where a candidate solution colored in blue does not satisfy this first criterion since $\exists \boldsymbol{x} \in \mathcal{A}\cup\mathcal{Q}$, $\hat{g}^i(\boldsymbol{x}) \le \hat{g}^i(\boldsymbol{y}^i)$ holds. For the i-th subproblem, the solution colored in blue may become a Pareto solution, but its convergence is estimated to be poor compared to the competitive solutions stored in $\mathcal{A}$.

- **The second criterion** is to select a well-diversified solution $\boldsymbol{y}^i$, which is estimated to be non-dominated by any other solution in $\mathcal{A}$. That is, $\boldsymbol{y}^i$ must be an estimated Pareto solution satisfying the condition;

$$\nexists \boldsymbol{x} \in \mathcal{A},\ \ \boldsymbol{x} \prec \boldsymbol{y^i}, \tag{8}$$

where a dominance relationship, $\boldsymbol{x} \prec \boldsymbol{y^i}$ is estimated as;

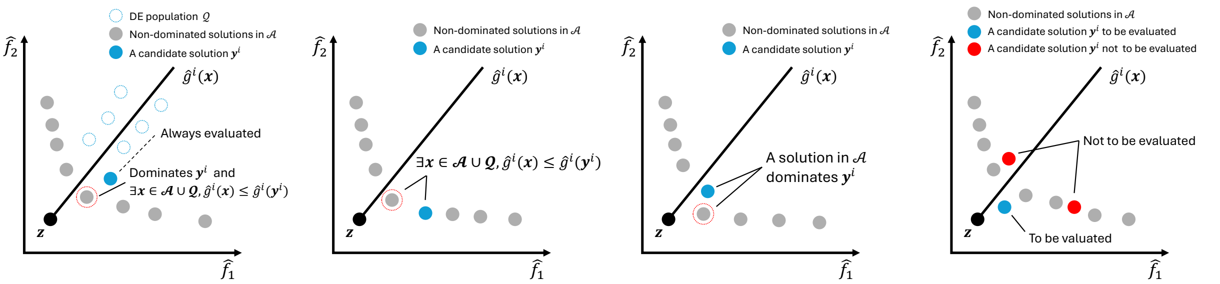
$$\begin{cases} \hat{f}_i(\boldsymbol{x}) \le \hat{f}_i(\boldsymbol{y}^i),\ \forall\, i \in \{1, \dots, M\} \\ \hat{f}_j(\boldsymbol{x}) < \hat{f}_j(\boldsymbol{y}^i),\ \exists\, j \in \{1, \dots, M\} \end{cases} \tag{9}$$

where $\hat{f}_i$ are approximated values of the objective values of f_i. This criterion contributes to improve both the diversity and convergence of the Pareto solutions. Fig.1(c) also illustrates a counterexample in which a candidate solution colored in blue is dominated by a competitive solution in $\mathcal{A}$, thereby failing to satisfy this second criterion. Note that approximation models of the objective functions are additionally required to predict dominance relationships, since SFA/DE constructs surrogate models only for the scalarization functions.

Accordingly, we design an infill criterion which incorporates the above two conditions, to identify more promising solutions that satisfy both conditions. Fig.1(d) shows an example in which a candidate solution $\boldsymbol{y}^i$ satisfies our two-step infill criterion. Our infill criterion encourages the selection of only the most promising solutions, to improve both the diversity and convergence.

3.2. Algorithm

SFA/DE-IC performs similarly to SFA/DE, except for the infill criterion. Algorithm 2 describes the pseudocode of SFA/DE-IC, where newly added procedures are underlined.



(a)SFA/DE

(b)A case in which the first criterion is unsatisfyied.

(c)A case in which the second criterion is unsatisfyied.

(d)A case in which both criteria are satisfyied.

Fig. 1: Graphical examples of candidate selection using different infill criteria

Given a set of pre-evaluated solutions $\mathcal{T}$, SFA/DE-IC initializes $\mathcal{A}$, $\mathcal{P}$, and $\boldsymbol{z}$ with the same manner as SFA/DE. As a main loop, for the i-th subproblem, it constructs an RBF model and then performs a DE-based search to obtain a candidate solution to be evaluated with the original objective functions, $\boldsymbol{y}^i$. As designed in SFA/DE, the candidate solution $\boldsymbol{y}^i$ with the minimum $\hat{g}^i$ value in $\mathcal{Q}$ is selected preliminarily to partially satisfy the first condition. Subsequently, the effectiveness of $\boldsymbol{y}^i$ is further validated using our infill criterion. To this end, SFA/DE-IC constructs M RBF models, $\hat{f}_j$, as approximation models of the objective functions, trained using all evaluated solutions in $\mathcal{A}$. Then, $\boldsymbol{y}^i$ is validated against the two conditions, i.e., Eq. (7) and Eq. (8). If $\boldsymbol{y}^i$ satisfies these conditions, it is evaluated with the original objective functions, and $\mathcal{A}$ and $\boldsymbol{z}$ are updated accordingly. Otherwise, SFA/DE-IC skips the evaluation of $\boldsymbol{y}^i$ as an undesirable solution, thereby saving the expensive FE. These processes are repeated for each subproblem[1].

4. EXPERIMENT

This section evaluates the effectiveness of SFA/DE-IC by comparing it with state-of-the-art SAMOEAs under highly expensive problem settings. All experiments reported in this paper were conducted using the evolutionary multiobjective optimization platform PlatEMO [17].

4.1. Experimental settings

4.1.1. Test problems

We used a DTLZ [18] test suite, which includes seven problem instances, DTLZ1-7. The number of objectives was set to $M = \{3, 7, 11\}$, and the number of decision variables was set to $D = \{50, 100\}$.

Regarding the availability of pre-evaluated solutions, this paper adopted a commonly used setting, assuming that $11D - 1$ pre-evaluated solutions, generated using the Latin Hypercube Sampling method, are available [12, 13, 19, 20]. The maximum number of FEs FE_{max} was set to 50 as a highly expensive setting, excluding those required to evaluate the pre-evaluated solutions.

[1] As an exceptional case, a candidate selection is randomly selected from $\{\boldsymbol{y}^i\}_{N}^{i=1}$ and then is evaluated if no solution is evaluated across all subproblems. However, this case did not occur in any of the experiments reported in this paper.

Algorithm 2 SFA/DE-IC

Input: Set of pre-evaluated solutions $\mathcal{T}$, Population size N, Maximum number of generations on DE-based search $w_{\max}$, Neighbor size T, Parameters of DE-based search F, CR
Output: Non-dominated solutions in $\mathcal{A}$
Set $\mathcal{B}(i)\ \forall i \in \{1, \dots, N\}$
Initialize an archive $\mathcal{A}$ as a duplicate of $\mathcal{T}$
Initialize a population $\mathcal{P}$ with N solutions selected from $\mathcal{T}$
Initialize $\boldsymbol{z}$ as $z_j = \min_{\boldsymbol{x} \in \mathcal{A}} f_j(\boldsymbol{x}) \forall j \in \{1, \dots, M\}$
while termination criteria are not satisfied **do**
  for $i = 1$ **to** N **do**
    Build a training dataset $\mathcal{D}^i$ with $\mathcal{A}$ by Eq. (5)
    Train an RBF model $\hat{g}^i$ with $\mathcal{D}^i$
    Initialize DE population $\mathcal{Q}$ by Eq. (6)
    for $w = 1$ **to** $w_{\max}$ **do**
      Set an offspring set $\mathcal{V}$ as an empty set
      for $j = 1$ **to** $|\mathcal{Q}|$ **do**
        Generate offspring solution $\boldsymbol{u}^j$ by applying crossover and mutation to $\boldsymbol{x}^j \in \mathcal{Q}$
        if $\hat{g}^i(\boldsymbol{u}^j) \le \hat{g}^i(\boldsymbol{x}^j)$ **then**
          Add $\boldsymbol{u}^j$ to $\mathcal{V}$
        else
          Add $\boldsymbol{x}^j$ to $\mathcal{V}$
        end if
      end for
      Update $\mathcal{Q}$ as $\mathcal{Q} \leftarrow \mathcal{V}$
    end for
    Set $\boldsymbol{y}^i$ as $\boldsymbol{y}^i \leftarrow \arg\min_{\boldsymbol{x} \in \mathcal{Q}} \hat{g}^i(\boldsymbol{x})$
    Train RBF models $\hat{f}_j\ \forall j \in \{1, \dots, M\}$ with $\mathcal{A}$
    if $\boldsymbol{y}^i$ satisfies both Eq. (7) and Eq. (8)
      Evaluate $\boldsymbol{y}^i$ with the original objective functions
      Replace $\boldsymbol{x}^j$ with $\boldsymbol{y}^i$ if $g(\boldsymbol{y}^i|\boldsymbol{\lambda}^j) \le g(\boldsymbol{x}^j|\boldsymbol{\lambda}^j)\ \forall j \in \mathcal{B}(i)$
      Add $\boldsymbol{y}^i$ to $\mathcal{A}$
      Update $\boldsymbol{z}$ as $z_j \leftarrow \min\{z_j, f_j(\boldsymbol{y}^i)\}\ \forall j \in \{1, \dots, M\}$
    end if
  end for
end while

4.1.2. Compared algorithms

The following seven state-of-the-art SAMOEAs were employed as compared algorithms: GCS-PSO [12], HeE-MOEA [19], HNN-MCEA/D [21], PIEA [14], XGBEA [13], MMRAEA [20], and SFA/DE [15]. Their parameter settings are summarized in Table 1. Note that SFA/DE-IC used the same parameters as SFA/DE as it requires no additional parameters. The default population size N was set to 50. For decomposition-based SAMOEAs, namely, HNN-MCEA/D, SFA/DE, and SFA/DE-IC, N was determined based on M using a two-layered approach [22], to obtain uniformly-distributed weight vectors.

Table 1: Parameter settings of compared algorithms.

Algorithm	Parameter setting
GCS-PSO [12]	$N = 50$, $g_{max} = 20$, $\mu = 5$, $div = 30$, $Pr = 0.8$, the parameters of the RF model are set as follows: features $= 0.8D$, samples $= 0.8\|DB\|$, trees $= 5$
HeE-MOEA [19]	$N = 50$, $k_m = 5$, $l = 11D - 1 + 25$, the population size and number of generations for NSGA-II are 50
HNN-MCEA/D [21]	$N = \{45, 35, 22\}$ for $M = \{3, 7, 11\}$, $\delta = 0.9$, $T = \lceil 0.1N \rceil$, $nr = 2$, $R_{max} = 10$, $l = 0.05$, $J = K = 20$, $bs = 25$, $iter_{\text{train}} = 50$, $F = 0.5$, $CR = 1.0$, $p_m = 1/D$, $\eta = 20$
MMRAEA [20]	$N = 50$, $\omega_{\max} = 20$, $\mu = 5$
PIEA [14]	$N = 50$, $\omega_{\max} = 20$, $\eta = 5$, $\tau = 20$, $F = 0.5$, $CR = 1.0$
XGBEA [13]	$N = 50$, $p_c = 1.0$, $p_m = 1/D$, $\eta_c = 20$, $\eta_m = 20$, $u = 5$, $\delta = 8$, $\eta = 0.1$, $iters = 200$, $\omega_{\max} = 20$, the sample selection probability is 0.6
SFA/DE [15]	$N = \{45, 35, 22\}$ for $M = \{3, 7, 11\}$, $T = \lceil 0.1N \rceil$, $F = 0.5$, $CR = 0.9$, $\omega_{\max} = 20$
SFA/DE-IC	The same settings as those used in SFA/DE were used.

The population of each algorithm was initialized with N solutions from $\mathcal{T}$. For all algorithms except for the three decomposition-based algorithms, their respective environmental selection mechanisms were used to determine the initial solutions. For the three decomposition-based algorithms, the top N solutions identified by non-dominated sorting were used to initialize the population.

4.1.3. Evaluation criteria

We used the modified Inverted Generational Distance (IGD^+) [23] as a performance metric. Smaller IGD^+ values indicate better algorithm performances. The IGD^+ values were computed using the non-dominated solutions identified from all evaluated solutions. We report the average IGD^+ values over 21 trials with different random seeds. The Wilcoxon rank-sum test was used to find significant differences, with a significance level of 0.05.

4.2. Results

Table 2 shows the average IGD^+ values for the DTLZ problems with $M = \{3, 7, 11\}$ and $D = \{50, 100\}$. Three symbols, "+" ,"−", and "≈" indicate that the IGD^+ value of a compared algorithm is significantly better than, significantly worse than, or comparable with that of SFA/DE-IC, respectively. The best IGD^+ value for each problem is highlighted in color. The average ranks and a summary of statistical results are provided at the bottom of each table. As shown in the table, the overall results indicate that SFA/DE-IC statistically outperformed the compared algorithms on many problem instances, even when the problem dimension was increased to 100. In particular, the IGD^+ values of SFA/DE-IC were statistically better than those of SFA/DE on seven problem instances for both $D = \{50, 100\}$. While the average rank of SFA/DE was worse than that of MMRAEA, SFA/DE-IC achieved the best average rank. These observations indicate that our infill criterion effectively adapted the SFA/DE framework to highly expensive MOPs with a limited budget of 50 FEs.

Fig.2 shows the convergence curves of IGD^+ values with respect to FEs for selected problem instances. The IGD^+ values of SFA/DE-IC tended to converge rapidly for many problem instances compared to those of the other algorithms. While SFA/DE achieved faster convergence on DTLZ1 with $\{M, D\} = \{11, 100\}$ and DTLZ6 with $\{7, 50\}$, the convergence speed slowed on DLTZ2 with $\{3, 100\}$. The convergence trends of the other algorithms varied depending on the problem instances. These observations suggest that SFA/DE-IC achieved better and more stable performance improvements, even with a limited budget of 50 FEs.

5. ANALYSIS

This section presents ablation studies to further validate the effectiveness of our infill criterion. Moreover, we investigate the substantive impact of SFA/DE-IC in terms of how many FEs SFA/DE-IC successfully reduced compared to standard optimizers.

5.1. Effectiveness of proposed infill criterion

To validate the effectiveness of our infill criterion, we here introduced three SFA/DE-IC variants, each employing a different infill criterion, as described below.

- **SFA/DE-IC-α**, which uses only the first criterion related to the convergence factor, that is, Eq. (7). This can be implemented by replacing Line 27 of Algorithm 2 with "**if** $\boldsymbol{y}^i$ satisfies Eq. (7)".
- **SFA/DE-IC-β**, which uses only the second criterion related to the estimated Pareto solution, that is, Eq. (8). This can be implemented by replacing Line 27 of Algorithm 2 with "**if** $\boldsymbol{y}^i$ satisfies Eq. (8)".
- **SFA/DE-IC-γ** does not use our infill criterion but instead randomly skips the evaluation of $\boldsymbol{y}^i$. Specifically, it selects $\boldsymbol{y}^i$ as having the minimum $\hat{g}^i$ value in $\mathcal{Q}$ (as designed in the original SFA/DE), and then evaluates $\boldsymbol{y}^i$ with a certain probability p_{skip}. This can be implemented by replacing Line 27 of Algorithm 2 with "**if** $rand < p_{skip}$", where $rand$ is a random value uniformly sampled from $[0, 1]$. We set $p_{skip} = 0.4$, which corresponds to the average skip rate observed in SFA/DE-IC across the experimental results.

The experimental settings used in Section 4 were applied. Table 3 shows the summary of the obtained statistical results (i.e., the counts of $+/-/\approx$), where "+","−", and "≈" indicate that the IGD^+ value of an SFA/DE-IC variant is significantly better than, significantly worse than, or comparable with that of SFA/DE-IC, respectively. For $D = 50$, the results showed slight improvements of SFA/DE-IC over SFA/DE-IC-α and SFA/DE-IC-β; SFA/DE-IC statistically outperformed there variants up to two problem instances. However, SFA/DE-IC still achieved the best average rank, and it sufficiently outperformed SFA/DE-IC-γ. For $D = 100$, the effectiveness of SFA/DE-IC over SFA/DE-IC-α and SFA/DE-IC-β was further highlighted. These observations suggest

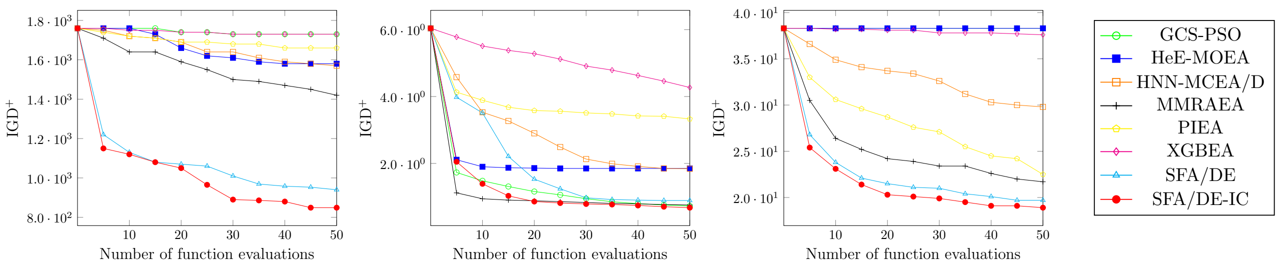
Table 2: Average IGD$^+$ values and average ranks on DTLZ problems (50FEs, 21trials).

(a) $D = 50$

Problem	M	GCS-PSO	HeE-MOEA	HNN-MCEA/D	MMRAEA	PIEA	XGBEA	SFA/DE	SFA/DE-IC
DTLZ1	3	1.164e+03 −	1.158e+03 −	1.089e+03 −	9.319e+02 −	9.333e+02 −	1.089e+03 −	5.709e+02 ≈	5.621e+02
	7	8.461e+02 −	8.554e+02 −	7.873e+02 −	7.710e+02 −	8.487e+02 −	8.532e+02 −	5.099e+02 ≈	4.978e+02
	11	7.350e+02 −	7.303e+02 −	6.514e+02 −	5.330e+02 −	7.115e+02 −	7.185e+02 −	4.364e+02 −	3.504e+02
DTLZ2	3	3.994e−01 ≈	6.160e−01 −	9.650e−01 −	3.929e−01 +	1.530e+00 −	1.295e+00 −	5.679e−01 −	4.396e−01
	7	7.109e−01 +	9.468e−01 +	1.178e+00 ≈	1.171e+00 ≈	1.672e+00 −	1.997e+00 −	1.421e+00 −	1.250e+00
	11	8.313e−01 +	1.098e+00 +	1.370e+00 −	1.275e+00 ≈	1.687e+00 −	2.025e+00 −	1.326e+00 ≈	1.231e+00
DTLZ3	3	3.623e+03 −	3.612e+03 −	2.810e+03 −	1.414e+03 −	2.082e+03 −	3.075e+03 −	1.194e+03 ≈	1.194e+03
	7	3.316e+03 −	3.315e+03 −	2.875e+03 −	1.318e+03 −	2.967e+03 −	3.281e+03 −	1.078e+03 ≈	1.043e+03
	11	2.918e+03 −	2.949e+03 −	2.060e+03 −	1.102e+03 −	2.677e+03 −	2.958e+03 −	9.522e+02 ≈	9.157e+02
DTLZ4	3	1.111e+00 −	8.169e−01 −	1.027e+00 −	9.256e−01 −	1.751e+00 −	1.848e+00 −	6.322e−01 −	5.986e−01
	7	1.266e+00 −	1.050e+00 −	1.058e+00 −	1.437e+00 −	1.838e+00 −	2.488e+00 −	9.037e−01 +	9.700e−01
	11	1.310e+00 −	1.086e+00 ≈	1.108e+00 +	1.641e+00 −	1.779e+00 −	2.505e+00 −	1.146e+00 ≈	1.120e+00
DTLZ5	3	3.282e−01 +	5.115e−01 −	9.610e−01 −	2.501e−01 +	1.415e+00 −	1.446e+00 −	4.997e−01 −	3.850e−01
	7	4.480e−01 +	5.571e−01 +	1.026e+00 −	7.721e−01 ≈	1.360e+00 −	1.714e+00 −	8.829e−01 −	6.965e−01
	11	4.686e−01 +	5.764e−01 +	9.101e−01 −	7.774e−01 ≈	1.201e+00 −	1.658e+00 −	8.393e−01 ≈	7.866e−01
DTLZ6	3	4.179e+01 −	4.177e+01 −	3.317e+01 −	1.855e+01 +	1.605e+01 +	3.565e+01 −	2.215e+01 ≈	2.108e+01
	7	3.829e+01 −	3.829e+01 −	2.981e+01 −	2.172e+01 −	2.253e+01 −	3.759e+01 −	1.965e+01 ≈	1.889e+01
	11	3.474e+01 −	3.466e+01 −	2.554e+01 −	1.846e+01 ≈	2.132e+01 −	3.454e+01 −	1.679e+01 ≈	1.691e+01
DTLZ7	3	9.445e+00 −	4.823e+00 +	9.018e+00 −	8.964e−01 +	6.570e+00 ≈	2.188e+00 +	7.285e+00 −	6.694e+00
	7	2.290e+01 −	2.020e+01 −	2.105e+01 −	1.016e+01 +	1.855e+01 ≈	6.509e+00 +	1.861e+01 ≈	1.731e+01
	11	3.581e+01 −	3.493e+01 −	3.262e+01 ≈	1.900e+01 +	2.832e+01 +	1.494e+01 +	3.049e+01 ≈	3.196e+01
+/−/≈		5/15/1	5/15/1	1/18/2	6/10/5	2/17/2	3/18/0	1/7/13	-
Avg. Rank		5.452	5.024	5.048	3.000	5.381	6.286	3.333	2.476

(b) $D = 100$

Problem	M	GCS-PSO	HeE-MOEA	HNN-MCEA/D	MMRAEA	PIEA	XGBEA	SFA/DE	SFA/DE-IC
DTLZ1	3	2.579e+03 −	2.588e+03 −	2.393e+03 −	2.364e+03 −	2.423e+03 −	2.539e+03 −	1.175e+03 −	1.157e+03
	7	1.864e+03 −	1.871e+03 −	1.739e+03 −	1.712e+03 −	1.864e+03 −	1.888e+03 −	1.096e+03 ≈	1.059e+03
	11	1.730e+03 −	1.575e+03 −	1.573e+03 −	1.419e+03 −	1.655e+03 −	1.734e+03 −	9.398e+02 ≈	8.488e+02
DTLZ2	3	7.267e−01 ≈	1.850e+00 −	1.844e+00 −	7.581e−01 −	3.332e+00 −	4.270e+00 −	8.911e−01 −	6.771e−01
	7	1.035e+00 +	2.354e+00 −	2.016e+00 ≈	1.745e+00 ≈	3.525e+00 −	5.484e+00 −	2.413e+00 −	1.980e+00
	11	1.246e+00 +	2.179e+00 −	2.110e+00 ≈	2.065e+00 ≈	3.609e+00 −	5.623e+00 −	2.114e+00 ≈	1.954e+00
DTLZ3	3	8.297e+03 −	8.381e+03 −	5.922e+03 −	3.136e+03 −	6.537e+03 −	7.591e+03 −	2.449e+03 −	2.443e+03
	7	7.888e+03 −	7.988e+03 −	6.118e+03 −	3.604e+03 −	7.186e+03 −	7.968e+03 −	2.341e+03 ≈	2.341e+03
	11	7.518e+03 −	7.537e+03 −	5.214e+03 −	2.738e+03 −	6.795e+03 −	7.596e+03 −	2.125e+03 ≈	2.147e+03
DTLZ4	3	1.834e+00 −	1.779e+00 −	1.845e+00 −	1.218e+00 −	3.537e+00 −	5.526e+00 −	8.827e−01 −	8.157e−01
	7	2.014e+00 −	2.046e+00 −	1.593e+00 −	1.949e+00 −	3.646e+00 −	5.979e+00 −	1.215e+00 ≈	1.426e+00
	11	2.136e+00 −	2.037e+00 −	1.611e+00 ≈	2.235e+00 −	3.519e+00 −	5.935e+00 −	1.523e+00 ≈	2.003e+00
DTLZ5	3	6.302e−01 ≈	1.782e+00 −	1.822e+00 −	5.853e−01 ≈	3.299e+00 −	4.378e+00 −	7.827e−01 −	6.050e−01
	7	9.453e−01 +	1.908e+00 −	2.040e+00 −	1.266e+00 ≈	3.253e+00 −	5.264e+00 −	1.862e+00 −	1.488e+00
	11	9.729e−01 +	1.774e+00 −	1.725e+00 ≈	1.585e+00 ≈	3.175e+00 −	5.319e+00 −	1.652e+00 ≈	1.557e+00
DTLZ6	3	8.630e+01 −	8.626e+01 −	6.767e+01 −	4.052e+01 +	5.482e+01 −	8.112e+01 −	4.574e+01 ≈	4.453e+01
	7	8.279e+01 −	8.278e+01 −	6.549e+01 −	4.922e+01 −	6.314e+01 −	8.228e+01 −	4.206e+01 ≈	4.332e+01
	11	7.922e+01 −	7.923e+01 −	5.848e+01 −	4.517e+01 −	6.194e+01 −	7.896e+01 −	4.052e+01 ≈	4.010e+01
DTLZ7	3	1.001e+01 −	6.680e+00 +	9.697e+00 −	2.561e+00 +	8.821e+00 ≈	4.955e+00 +	9.193e+00 ≈	8.963e+00
	7	2.430e+01 −	2.214e+01 ≈	2.290e+01 ≈	1.608e+01 +	2.315e+01 ≈	1.339e+01 +	2.253e+01 ≈	2.189e+01
	11	3.868e+01 −	3.791e+01 ≈	3.694e+01 ≈	2.762e+01 +	3.639e+01 ≈	2.705e+01 +	3.708e+01 ≈	3.636e+01
+/−/≈		4/15/2	1/18/2	0/15/6	4/12/5	0/18/3	3/18/0	0/7/14	-
Avg. Rank		5.238	6.000	4.619	2.714	5.810	6.524	3.048	2.048



(a)DTLZ1 with $M = 11, D = 100$ (b)DTLZ2 with $M = 3, D = 100$ (c)DTLZ6 with $M = 7, D = 50$

Fig. 2: Convergence curves of average IGD$^+$ values for selected problem instances.

Table 3: Summary of statistical results of the SFA/DE-IC variants ($+/-/\approx$).

(a)$D = 50$

	SFA/DE-IC-α	SFA/DE-IC-β	SFA/DE-IC-γ	SFA/DE-IC
$+/-/\approx$	0/2/19	1/2/18	0/7/14	-
Avg. Rank	2.643	2.333	3.238	1.786

(b)$D = 100$

	SFA/DE-IC-α	SFA/DE-IC-β	SFA/DE-IC-γ	SFA/DE-IC
$+/-/\approx$	0/4/17	0/3/18	0/5/16	-
Avg. Rank	2.857	2.286	2.857	2.000

Table 4: Summary of statistical results of MOEA/D and random search compared to SFA/DE-IC ($+/-/\approx$).

D	MOEA/D	random search
50	0/20/1	0/21/0
100	0/19/2	0/21/0

that the synergy between our two criteria helps to select more promising solutions, thereby improving the performance of SFA/DE-IC.

5.2. Comparison to standard optimizers

A common motivation for SAMOEAs is to reduce the number of FEs required by standard (model-free) optimizers, such as MOEAs. We compare SFA/DE-IC with two standard optimizers, MOEA/D and a random search algorithm to assess its substantive impact. MOEA/D is the base optimizer used in both SFA/DE and SFA/DE-IC, and its population was initialized using the same procedure as in SFA/DE-IC with the set of pre-evaluated solutions $\mathcal{T}$. The random search was implemented to randomly generate FE_{max} solutions. The experimental settings used in Section 4 were applied.

Table 4 shows the counts of ($+/-/\approx$), where "$+$","$-$", and "$\approx$" indicate that the IGD$^+$ value of a model-free optimizer is significantly better than, significantly worse than, or comparable with that of SFA/DE-IC, respectively. As shown in this table, SFA/DE-IC statistically outperformed the random search for all 21 problem instances. Moreover, it sufficiently outperformed MOEA/D on at least 19 problem instances.

Fig. 3 shows the convergence curves of the average IGD$^+$ values obtained by MOEA/D and SFA/DE-IC for selected problem instances. Based on the average IGD$^+$ values of SFA/DE-IC at 50 FEs, SFA/DE-IC reduces the number of FEs required by MOEA/D by at least 97.3%. These results suggest the significant effectiveness of SFA/DE-IC in reducing the number of expensive FEs.

6. CONCLUSION

This paper proposed an extension of SFA/DE tailored to highly expensive MOPs, where the number of expensive FEs is restricted to several dozen. We developed a novel infill criterion that aims to identify more promising solutions by rigorously evaluating their contributions to both convergence and diversity, and integrated it into the SFA/DE framework. Experimental results demonstrate that the proposed algorithm significantly outperforms existing SAMOEAs under a strict budget of 50 FEs. Furthermore, the proposed algorithm, SFA/DE-IC, skips the evaluation of undesirable solutions that do not satisfy our condition, thereby saving expensive FEs. Experimental results showed that SFA/DE-IC clearly outperforms the state-of-the-art algorithms under a limited budget of 50 FEs on 50- and 100-dimensional benchmark problems. Moreover, our analysis reveals that the proposed infill criterion improves the sampling efficiency of SFA/DE by enabling it to select more effective solutions.

In future work, we will investigate the effectiveness of SFA/DE-IC on real-world problems, such as Human-Powered Aircraft Design problems (HPA) [3] and Car Structure Design Optimization Problem (CSDOP) [24]. Moreover, we also aim to improve the quality of surrogate models by effectively using the training samples. In particular, ensemble approaches that combine multiple surrogate models [20, 25], and dimensionality reduction-based surrogate models [19, 26], represent a promising direction for improving model robustness and accuracy.

ACKNOWLEDGMENT

This paper is based on results obtained from a project, JPNP20004, subsidized by New Energy and Industrial Technology Development Organization (NEDO).

REFERENCES

[1] Alan Díaz-Manríquez, Gregorio Toscano, Jose Hugo Barron-Zambrano, and Edgar Tello-Leal. A Review of Surrogate Assisted Multiobjective Evolutionary Algorithms. *Comput. Intell. Neurosci.*, 2016(1):9420460, 2016.

[2] Timothy W Simpson, Andrew J Booker, Dipankar Ghosh, Anthony A Giunta, Patrick N Koch, and R-J Yang. Approximation methods in multidisciplinary analysis and optimization: a panel discussion. *Struct. Multidiscip. Optim*, 27:302–313, 2004.

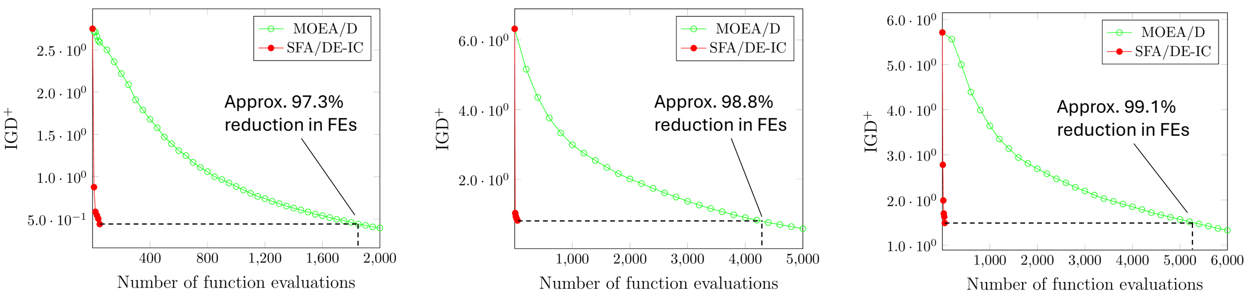
[3] Nobuo Namura. Single and multi-objective optimization benchmark problems focusing on human-powered aircraft design. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 195–210. Springer, 2025.

[4] Nicholas Aksamit, Jinqiang Hou, Yifeng Li, and Beatrice Ombuki-Berman. Integrating transformers and many-objective optimization for drug design. *BMC BMC Bioinf.*, 25(1):208, 2024.

[5] Akira Oyama, Takehisa Kohira, Hiromasa Kemmotsu, Tomoaki Tatsukawa, and Takeshi Watanabe. Simultaneous structure design optimization of multiple car models using the K computer. In *2017 IEEE Symp. Ser. Comput. Intell. (SSCI)*, pages 1–4. IEEE, 2017.

[6] Jiao Liu, Yong Wang, Guangyong Sun, and Tong Pang. Solving Highly Expensive Optimization Problems via Evolutionary Expected Improvement. *IEEE Trans. Syst. Man Cybern.: Syst.*, 53(8):4843–4855, 2023.

[7] Yaochu Jin. Surrogate-assisted evolutionary com-



(a)DTLZ2 with $M = 3, D = 50$ (b)DTLZ4 with $M = 3, D = 100$ (c)DTLZ5 with $M = 7, D = 100$

Fig. 3: Convergence curves of average IGD$^+$ values obtained by MOEA/D and SFA/DE-IC for selected problem instances.

putation: Recent advances and future challenges. *Swarm Evol. Comput.*, 1(2):61–70, 2011.

[8] Christopher Williams and Carl Rasmussen. Gaussian Processes for Regression. *Adv. Neural Inf. Process. Syst.*, 8, 1995.

[9] David S Broomhead and David Lowe. Radial Basis Functions, Multi-Variable Functional Interpolation and Adaptive Networks. Technical report, Royal Signals and Radar Establishment Malvern (United Kingdom), 1988.

[10] Del Nelson and Jun Wang. Introduction to artificial neural systems, 1992.

[11] Takumi Sonoda and Masaya Nakata. Multiple Classifiers-Assisted Evolutionary Algorithm Based on Decomposition for High-Dimensional Multiobjective Problems. *IEEE Trans. Evol. Comput.*, 26(6):1581–1595, 2022.

[12] Qi-Te Yang, Zhi-Hui Zhan, Xiao-Fang Liu, Jian-Yu Li, and Jun Zhang. Grid Classification-Based Surrogate-Assisted Particle Swarm Optimization for Expensive Multiobjective Optimization. *IEEE Trans. Evol. Comput.*, 2023.

[13] Feiqiao Mao, Ming Chen, Kaihang Zhong, Jiyu Zeng, and Zhengping Liang. An XGBoost-assisted evolutionary algorithm for expensive multiobjective optimization problems. *Inf. Sci.*, 666:120449, 2024.

[14] Yang Li, Weigang Li, Songtao Li, and Yuntao Zhao. A performance indicator-based evolutionary algorithm for expensive high-dimensional multi-/many-objective optimization. *Inf. Sci.*, 678:121045, 2024.

[15] Yuma Horaguchi, Kei Nishihara, and Masaya Nakata. Evolutionary multiobjective optimization assisted by scalarization function approximation for high-dimensional expensive problems. *Swarm Evol. Comput.*, 86:101516, 2024.

[16] Qingfu Zhang and Hui Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Trans. Evol. Comput.*, 11(6):712–731, 2007.

[17] Ye Tian, Ran Cheng, Xingyi Zhang, and Yaochu Jin. PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization. *IEEE Comput. Intell. Mag.*, 12(4):73–87, 2017.

[18] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable Test Problems for Evolutionary Multiobjective Optimization. In *Evolutionary multiobjective optimization: theoretical advances and applications*, pages 105–145. Springer, 2005.

[19] Dan Guo, Yaochu Jin, Jinliang Ding, and Tianyou Chai. Heterogeneous Ensemble-Based Infill Criterion for Evolutionary Multiobjective Optimization of Expensive Problems. *IEEE Trans. Cybern.*, 49(3):1012–1025, 2018.

[20] Jiangtao Shen, Xinjing Wang, Ruixuan He, Ye Tian, Wenxin Wang, Peng Wang, and Zhiwen Wen. Optimization of high-dimensional expensive multi-objective problems using multi-mode radial basis functions. *Complex Intell. Syst.*, 11(2):147, 2025.

[21] Bingdong Li, Yanting Yang, Wenjing Hong, Peng Yang, and Aimin Zhou. Hyperbolic Neural Network Based Preselection for Expensive Multi-Objective Optimization. *IEEE Trans. Evol. Comput.*, 2024.

[22] Kalyanmoy Deb and Himanshu Jain. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Trans. Evol. Comput.*, 18(4):577–601, 2014.

[23] Hisao Ishibuchi, Hiroyuki Masuda, Yuki Tanigaki, and Yusuke Nojima. Modified Distance Calculation in Generational Distance and Inverted Generational Distance. In *Evol. Multi-Criter. Optim. (EMO)*, pages 110–125. Springer, 2015.

[24] Takehisa Kohira, Hiromasa Kemmotsu, Oyama Akira, and Tomoaki Tatsukawa. Proposal of benchmark problem based on real-world car structure design optimization. In *Proc. Genet Evol. Comput. Conf.*, pages 183–184, 2018.

[25] Yuma Horaguchi and Masaya Nakata. A Dual Surrogate-Based Evolutionary Algorithm for High-Dimensional Expensive Multiobjective Optimization Problems. In *2024 IEEE Congr. Evol. Comput (CEC)*, pages 1–8. IEEE, 2024.

[26] Yuma Horaguchi and Masaya Nakata. High-Dimensional Expensive Optimization by Classification-based Multiobjective Evolutionary Algorithm with Dimensionality Reduction. In *2023 62nd Annu. Conf. Soc. Instrum. Control Eng. (SICE)*, pages 1535–1542. IEEE, 2023.