Region Division and Merging-based Multiobjective Optimization for Multimodal Problems

Jinxin Zhang, Billy Chiu, Jing-Yu Ji, *Member, IEEE,* Liping Liang, Jun Zhang, *Fellow, IEEE,* and Sam Kwong, *Fellow, IEEE*

***Abstract*—Recently, considerable research efforts have been directed toward multimodal optimization problems in consumer electronics. Given the existence of multiple equally significant solutions applicable to diverse scenarios, simultaneously locating these optima is essential. Utilizing multiobjective optimization presents a promising avenue for addressing such problems, provided an effective transformation from multimodal optimization to multiobjective optimization is established. To this end, this study proposes an enhanced tri-objective transformation framework based on a specifically designed region division and merging strategy. Three objective functions are constructed by incorporating objective conflict and niching principles, thus enabling multiobjective optimization techniques to efficiently locate multiple optimal solutions. Simultaneously, the proposed region division and merging strategy facilitates the transformation process. Initially, the decision space is partitioned into numerous small tiles during the region division phase. Subsequently, these tiles are progressively merged layer by layer during the merging phase, eventually restoring the original decision space. Consequently, the given optimization problem is decomposed into a series of simplified tri-objective optimization subproblems, facilitating a smooth transition from exploration of potential regions to exploitation of promising candidate solutions. Experiments conducted on 20 benchmark multimodal optimization problems demonstrate that the proposed method achieves superior performance compared to 14 state-of-the-art algorithms in terms of simultaneously identifying multiple optimal solutions.**

***Index Terms*—Region division and merging, transformation, multiple optimal solutions, multiobjective optimization, differential evolution**

I. Introduction

Many consumer electronics applications, such as the autonomous path planning [1], [2] and traveling salesman problems [3], [4], exhibit the coexistence of multiple optimal (or near-optimal) solutions. For example, consumers may prefer to have multiple optimal solutions available before making purchasing decisions for electronic products [5]. Offering representative solutions to support consumers' final decision-making can be framed as a multimodal optimization problem (MMOP), which is formulated as follows[1] [6], [7]:

$$\begin{array}{ll} \text{minimize} & g(\mathbf{x}), \\ \text{subeject to} & \mathbf{x} = (x_1, \ldots, x_D) \in X, \end{array} \tag{1}$$

where $g(\mathbf{x})$ is the objective function, D denotes the number of decision variables, $\mathbf{x}$ is a decision vector in the decision space X, and X can be defined as follows:

$$X = \textstyle\prod_{i=1}^{D} [l_i, u_i], \tag{2}$$

where l_i and u_i are the lower and upper bounds of x_i, respectively. Ideally, an algorithm should simultaneously identify all global optima in a single run, providing consumers with well-informed choices for final decision-making.

While population-based evolutionary algorithms (EAs) have been successfully applied to optimization problems in consumer electronics [8]–[12], they predominantly focus on modeled search problems with a similar formulation but only a single optimal solution. MMOPs in consumer electronics, however, pose distinct challenges. On the one hand, the number of solutions found must be as large as possible to accommodate consumers' diverse preferences. On the other hand, the quality of the identified solutions must meet the practical demands of electronics. Consequently, additional multimodality-specific mechanisms need to be specifically designed to address MMOPs in consumer electronics.

Niching [13]–[15], one of the most popular methods for handling multimodality, has been widely used in various EAs. During the evolutionary progress, niching achieves two goals [13]. First, it adaptively partitions the entire population into different subpopulations, and second, it maintains subpopulation diversity to exploit different promising candidate solutions. Various niching strategies have been developed, such as clearing [16], crowding [17], fitness sharing [14], [18], and clustering [19]. In the early studies, the performance of these methods largely relies on the niching threshold parameter. However, different MMOPs typically have different numbers of local and global optima, making the threshold parameter problem-dependent and/or user-defined [13]. Thus, these early techniques commonly face the issue of parameter sensitivity. Recent studies have focused on parameter-free or parameter-insensitive improvements for niching, such as adaptive estimation distribution [20], [21], topological speciation [22], and

This research was supported by the Scientific Research Foundation for the Phase III Construction of a High-Level University for Youth Scholars at Shenzhen University (Grant No. 000001032933), and by the Interdisciplinary Team Research Project of the College of Management, Shenzhen University (Grant No. 20240408). *(Corresponding author: Jing-Yu Ji.)*

J. Zhang and J.-Y. Ji are with the College of Management, Shenzhen University, Shenzhen, China. (email: im@jijingyu.com)

B. Chiu and S. Kwong are with the School of Data Science, Lingnan University, Hong Kong.

L. Liang is with the Department of Operations and Risk Management, Lingnan University, Hong Kong.

J. Zhang is with the College of Artificial Intelligence, Nankai University, Tianjin 300350, China, and also with Hanyang University, Ansan 15588, South Korea.

[1] The maximization problem can be equivalently transformed into a minimization problem by multiplying by a negative value.

affinity propagation clustering [23]. Leveraging adaptive and self-adaptive strategies, these new niching methods can be less sensitive or even insensitive to user-defined parameters.

As an alternative to niching methods, multimodality-specific reproduction and updating operators [24]–[26] have been proposed to recognize multiple optimal solutions using variable information. Unlike conventional SOPs, which have a single optimal solution that exclusively achieves the minimum objective function value, information solely from the objective function of a given MMOP is insufficient to locate multiple optima with the same objective value. To distinguish different optima, the original reproduction and updating operations have been redesigned to consider information from individual variables. For example, sub-decision space information is used in the updating operator to locally guide different individuals toward distinct directions [25]. The neighborhood mutation [27] utilizes local information provided by neighboring individuals to explore different decision areas.

Recently, some efforts [6], [7], [28]–[30] have been made to solve MMOPs using multiobjective optimization, which was originally proposed for multiobjective optimization problems (MOPs). The similarity between MMOPs and MOPs is that both involve a set of optimal solutions that should be located in a single run. However, a key difference is that MMOPs have only one objective function and thus cannot be solved by multiobjective optimization directly. Therefore, transforming a given MMOP into an MOP is essential. In [28], [29], [31], a bi-objective-based transformation has been proposed. The first objective is derived from the given MMOP, and the second objective, serving as a diversity indicator, is based on gradient information. Furthermore, to fully leverage multiobjective optimization, a novel bi-objective transformation based on objective conflict has been designed [6]. This approach rigorously converts the multiple optimal solutions of a given MMOP into the Pareto-optimal solutions of the transformed MOP, thus satisfying the prerequisites for applying multiobjective optimization.

Using multiobjective optimization to solve MMOPs reduces the need to design new algorithms [7], as existing multiobjective optimization techniques can be suitably adapted through appropriate modifications. Nevertheless, the successful application of multiobjective optimization critically relies on an efficient transformation of MMOPs into MOPs. In this study, we propose a region division and merging (RDM) strategy to facilitate a tri-objective transformation. Accordingly, we develop a tri-objective differential evolution (TriDE) algorithm. Unlike most prevailing approaches [7], [28], [29], [31] that transform an MMOP into a fixed bi-objective optimization problem (BOP), we partition the decision space into multiple tiles (sub-decision spaces), thus transforming the original MMOP into a set of simpler tri-objective optimization problems (TOPs). Subsequently, the proposed TriDE algorithm systematically explores each tile during the initial and intermediate phases to identify potential attraction basins likely to contain promising solutions. When the merging strategy is activated, pairs of these tiles are progressively integrated. Finally, promising candidate solutions identified in earlier stages undergo further exploitation to yield high-quality final solutions. The main advantages of the proposed TriDE algorithm are summarized as follows:

1) Three objective functions are carefully constructed in TriDE for the transformation of MMOPs. The first two objectives inherit the essential objective conflicts from bi-objective transformation to fulfill multiobjective optimization requirements, while the third objective is developed from niching strategy to maintain population diversity. Thus, TriDE effectively leverages both multiobjective optimization and niching to identify multiple optimal solutions.
2) An effective RDM strategy is introduced to facilitate the transformation of MMOPs into MOPs. Initially, the entire decision space is segmented into numerous smaller tiles, which are gradually combined back into larger tiles during evolution. Consequently, distinct search layers are formed across different stages. Earlier layers, comprising a greater number of smaller tiles, enable extensive exploration of potential regions, whereas later layers, consisting of fewer but larger tiles, allow focused exploitation of promising candidate solutions. This mechanism ensures a balanced and smooth transition from exploration to exploitation within TriDE.
3) The decision space is efficiently evaluated and partitioned using sampling and clustering techniques. Consequently, TriDE can subdivide decision spaces of arbitrary dimensions into a designated number of sub-decision spaces without requiring any *a priori* knowledge, thereby significantly alleviating the curse of dimensionality encountered in high-dimensional optimization scenarios.

The remainder of this paper is structured as follows. Section II provides the preliminaries. Section III presents a comprehensive description of the proposed TriDE method. Parameter settings and experimental results are detailed in Section IV. Finally, Section V concludes the paper.

II. PRELIMINARIES

A. Mutiobjective Optimization Problems

Without loss of generality, a multiobjective optimization problem (MOP) can be formulated as follows:

$$\begin{aligned} \text{minimize} \quad & F(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_M(\mathbf{x})), \\ \text{subeject to} \quad & \mathbf{x} \in X, \end{aligned} \tag{3}$$

where $F(\mathbf{x})$ denotes an objective vector in the objective space Y where $Y \subseteq \Re^M$, and M denotes the number of objectives.

Since an MOP often involves at least two conflicting objectives, it is unlikely that a solution in X exists that minimizes all objectives simultaneously. Instead, a set of solutions can be maintained to balance different objectives. The goal of multiobjective optimization is to find the best trade-offs, known as the Pareto-optimal solutions.

In multiobjective optimization, an important concept is Pareto dominance. Given two decision vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ in X, $\mathbf{x}_1$ is said to Pareto-dominate $\mathbf{x}_2$, denoted by $\mathbf{x}_1 \prec \mathbf{x}_2$, if for all $i \in \{1, \dots, M\}$ such that $f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2)$ and there exists $i \in \{1, \dots, M\}$ such that $f_i(\mathbf{x}_1) < f_i(\mathbf{x}_2)$. Furthermore, if no $\mathbf{x}_2$ exists such that $\mathbf{x}_2 \prec \mathbf{x}_1$, $\mathbf{x}_1$ is called a Pareto-optimal solution. The set of all Pareto-optimal solutions in the objective space is known as the Pareto front.

B. Fitness Sharing

Niching [32] has been one of the most commonly used techniques in population-based EAs to handle multimodality and identify multiple optimal solutions. By maintaining a diverse set of potential solutions, the evolutionary population can simultaneously search different areas of the solution space. Among various niching-based methods, fitness sharing [13] is one of the most popular. Since the proposed approach, TriDE, utilizes fitness sharing to construct a density function, we briefly describe this technique here.

Given an individual $\mathbf{x}$ with fitness $g(\mathbf{x})$, its shared fitness $g_s(\mathbf{x})$ is calculated as follows:

$$g_s(\mathbf{x}) = \frac{g(\mathbf{x})}{m_c(\mathbf{x})}, \tag{4}$$

where $m_c(\mathbf{x})$ is the niche count, representing the number of individuals with whom the fitness $g(\mathbf{x})$ is shared. The niche count is calculated as follows:

$$m_c(\mathbf{x}) = \sum_{i=1}^{P_S} \max(1 - \frac{\|\mathbf{x} - \mathbf{x}_i\|}{\sigma}, 0), \tag{5}$$

where P_S denotes the population size, $\| \cdot \|$ represents the Euclidean distance, and σ denotes the niche radius.

The principle of fitness sharing is to reduce the number of similar individuals within a given niche. By discouraging the presence of highly similar individuals, fitness sharing promotes population diversity, enabling the evolutionary search to effectively exploit different niches.

C. Multiobjective Optimization for MMOPs

Multiobjective optimization aims to find the best trade-offs between objectives. Since MMOPs have only one objective function, multiobjective optimization cannot be applied directly. Therefore, a well-designed transformation from an MMOP to an MOP is usually required. To achieve this, additional objective functions must be constructed. Current transformations can generally be categorized into two classes: 1) density indicator-based and 2) objective conflict-based transformations.

The density indicator as an additional objective function was proposed in [29] and further developed in [28], [33]. In this approach, a given MMOP is transformed into a BOP as follows:

$$\begin{aligned} &\text{minimize} \quad && F(\mathbf{x}) = (g(\mathbf{x}), d(\mathbf{x})), \\ &\text{subeject to} \quad && \mathbf{x} \in X, \end{aligned} \tag{6}$$

where $d(\mathbf{x})$ denotes the density function, and $g(\mathbf{x})$ and X have the same formulations as in (1). Typically, $d(\mathbf{x})$ uses distance-related information to reflect the density level around $\mathbf{x}$. When $d(\mathbf{x})$ is minimized, individuals occupying sparse areas have more opportunities for exploration and exploitation, helping the population avoid converging toward a single dominant region.

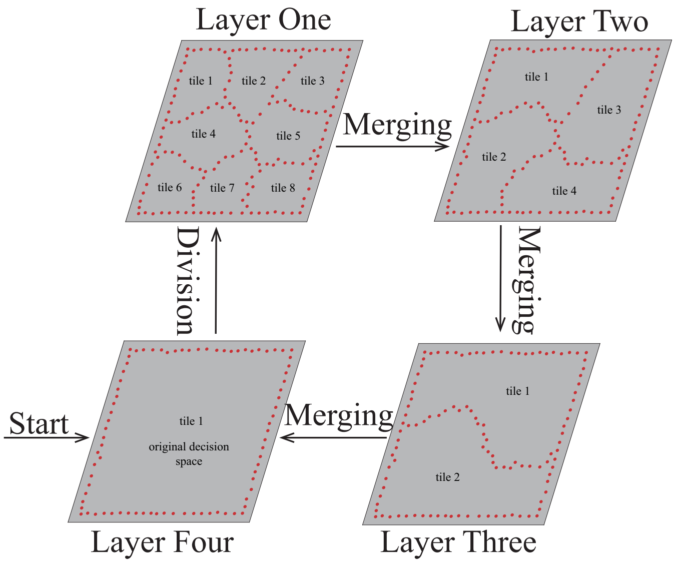


Fig. 1. Illustration of the RDM strategy. The decision space is initially divided into eight small tiles in the first layer, and these tiles subsequently merge back into the original space in the last layer.

Considering the prerequisite for the coexistence of Pareto-optimal solutions, a transformation involving two conflicting objectives was first proposed to find multiple solutions of nonlinear equation systems in [30]. Later, a more robust and rigorous transformation was developed to solve MMOPs in [6]. The general framework of such transformations can be summarized as follows:

$$\text{minimize} \begin{cases} f_1(\mathbf{x}) = \alpha(\mathbf{x}) + \beta(x) \\ f_2(\mathbf{x}) = 1 - \alpha(\mathbf{x}) + \beta(\mathbf{x}) \end{cases}, \tag{7}$$

where $\alpha(\mathbf{x})$ and $\beta(\mathbf{x})$ are distance-related and $g(\mathbf{x})$-related functions, respectively. In (7), $\alpha(\mathbf{x})$ serves two purposes. First, it introduces a complete conflict between $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$, enabling multiobjective optimization to identify the transformed optima. Second, it ensures that all optimal solutions of an MMOP can be converted into Pareto-optimal solutions.

III. Proposed Algorithm

A. Motivation

In terms of similarity, both MOPs and MMOPs involve multiple equally important optima that need to be located in a single trial. This similarity suggests a potential advantage in using multiobjective optimization to locate multiple optima. However, there are notable differences between MOPs and MMOPs. The key distinction is that the multiple optima of a given MMOP do not exhibit a regularity property as strong as the Pareto-optimal solutions of a standard MOP [7]. Any two optima of an MMOP have minimal or no correlation, as these optima are sparsely and discretely distributed across different attraction basins. Due to these differences, if multiobjective optimization is applied to MMOPs, some modifications are necessary [7].

Motivated by the aforementioned similarities and differences, an RDM strategy is proposed to better facilitate the transformation from MMOPs to MOPs. A brief illustration is provided in Fig. 1. In the first layer, the entire decision space is divided into a number of tiles, with each tile containing a population that undergoes multiobjective optimization. Although the tiles do not exhibit a regularity property like the Pareto-optimal solutions, each divided tile has an opportunity to be explored. As the evolutionary process progresses, these small tiles gradually merge into larger ones. In Fig. 1, smaller tiles, along with their corresponding subpopulations in the first two layers, become larger in the last two layers. Consequently, promising candidate solutions identified in earlier stages can be further refined with a larger population size to obtain high-accuracy solutions.

B. The Improved Tri-objective Transformation

In TriDE, a given MMOP is transformed into a tri-objective optimization problem (TOP), formulated as follows:

$$\begin{array}{ll} \text{minimize} & \begin{cases} f_1(\mathbf{x}) = \alpha(\mathbf{x}) + \beta(\mathbf{x}) \\ f_2(\mathbf{x}) = 1 - \alpha(\mathbf{x}) + \beta(\mathbf{x}) \\ f_3(\mathbf{x}) = m_c(\mathbf{x}) + \beta(\mathbf{x}) \end{cases}, \\ \text{subeject to} & \mathbf{x} = (x_1, \ldots, x_D) \in X, \end{array} \tag{8}$$

First, inspired by the successful applications in [6], [30], the transformation based on objective conflict is incorporated. Second, niching is another widely used method to enhance population diversity and identify multiple optimal solutions. Thus, the above tri-objective transformation is developed. Borrowing from (7), the distance-related function $\alpha(\mathbf{x})$ and $g(\mathbf{x})$-related function $\beta(\mathbf{x})$ in (8) are constructed as follows:

$$\alpha(\mathbf{x}) = \frac{\sum_{j=1}^{P_S} \|(\mathbf{x} - \mathbf{r}_j)\| - \min\limits_{t=1,\ldots,P_S} \sum_{j=1}^{P_S} \|(\mathbf{x}_t - \mathbf{r}_j)\|}{\max\limits_{t=1,\ldots,P_S} \sum_{j=1}^{P_S} \|(\mathbf{x}_t - \mathbf{r}_j)\| - \min\limits_{t=1,\ldots,P_S} \sum_{j=1}^{P_S} \|(\mathbf{x}_t - \mathbf{r}_j)\|}, \tag{9}$$

$$\beta(\mathbf{x}) = \xi \cdot \frac{g(\mathbf{x}) - \min\limits_{t=1,\ldots,P_S} g(\mathbf{x}_t)}{\max\limits_{t=1,\ldots,P_S} g(\mathbf{x}_t) - \min\limits_{t=1,\ldots,P_S} g(\mathbf{x}_t)},$$
$$\xi = 40D(CurrentFEs/MaxFEs)^3, \tag{10}$$

where $\mathbf{r} = (r_1, \ldots, r_D)$ represents a sample point randomly initialized within the decision space X, $\| \cdot \|$ denotes the Euclidean distance, P_S is the number of sample size (also the population size), and ξ is a scaling factor borrowed from [6]. Here, $CurrentFEs$ denotes the current number of fitness evaluations used.

It can be shown (Appendix A in the supplementary material) that $f_1(\mathbf{x})$ completely conflicts with $f_2(\mathbf{x})$, resulting in a mapping from the multiple optima of the original MMOP to the Pareto-optimal solutions of the transformed TOP. An example is depicted in Fig. S.1 (supplementary material). In Fig. S.1(a), six optimal solutions are located in distinct attraction basins with varying fitness landscapes. These six optimal solutions are marked as black points in the decision space. Correspondingly, any attraction basin containing an optimal solution (peak) is colored dark crimson, while basins without an optimal solution are shown as lighter areas. From (8) and (9), it can be theoretically deduced that the relationship between $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ is $f_2(\mathbf{x}) = 1 - f_1(\mathbf{x}) + 2\beta(\mathbf{x})$. Specifically, if $\mathbf{x}$ is an optimal solution, then $\beta(\mathbf{x}) = 0$. Fig. S.1(b) illustrates that the transformed six optimal solutions fall onto the line segment $f_2(\mathbf{x}) = 1 - f_1(\mathbf{x})$, with each solution being nondominated.

In contrast to constructing two conflicting objectives, the purpose of $f_3(\mathbf{x})$ is to distribute the transformed Pareto-optimal solutions more sparsely on the PF. For instance, in Fig. S.1(b), although points A and B are nondominated, their transformed positions are so close that it is challenging for the evolutionary search to precisely distinguish them simultaneously. To address this issue, the third objective $f_3(\mathbf{x})$, derived from the fitness sharing of the niche count $m_c(\mathbf{x})$, is constructed. However, $f_3(\mathbf{x})$ does not provide Pareto dominance in relation to either $f_1(\mathbf{x})$ or $f_2(\mathbf{x})$. As shown in Fig. S.1(c), when only $f_1(\mathbf{x})$ and $f_3(\mathbf{x})$ are considered, the relative distance between A and B is larger than that in Fig. S.1(b), yet Pareto dominance is absent. When all three objectives, $f_1(\mathbf{x})$, $f_2(\mathbf{x})$, and $f_3(\mathbf{x})$, are considered, the red points are transformed into optimal solutions in the three-dimensional objective space shown in Fig. S.1(d). Points A and B on the PF formed by $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ are the projection points of A' and B', indicating that A' and B' remain nondominated. Overall, compared with the transformation based solely on $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$, the nondominated solutions transformed by $f_1(\mathbf{x})$, $f_2(\mathbf{x})$, and $f_3(\mathbf{x})$ exhibit a greater degree of distinctness, enhancing solution identification.

C. RDM Strategy

Although the multiple optima of an original MMOP are transformed into the Pareto-optimal solutions of (8), a regularity property that could stimulate evolutionary search to locate additional nondominated solutions does not exist. Constructing such regularity is challenging. Therefore, an RDM strategy is proposed from an alternative perspective to enhance the evolutionary search capability.

1) Region Division: Often, the multiple optima of a given MMOP are sparsely distributed across distinct attraction basins, with each optimal solution effectively serving as an isolated point. To enhance the evolutionary search capability, we divide the decision space into multiple sub-decision spaces and then apply multiobjective optimization within each divided sub-decision space. By enforcing search within each of these divided spaces, bias toward any specific promising candidate solution can be largely mitigated, providing greater opportunity to explore potential attraction basins. In this study, we use the K-means clustering method to partition the decision space. The pseudo-code for region division is presented in **Algorithm 1**.

In the region division, Latin hypercube sampling (LHS) [34], [35] is employed to generate P_S sample points within the decision space X. After applying the K-means clustering method, these P_S sample points are grouped into N_T clusters; that is, the kth tile consists of p_s^k sample points, with $P_S = \sum_{k=1}^{N_T} p_s^k$. Here $p_s^k, k = 1, \ldots, N_T$ is determined automatically by the K-means clustering method. Consequently, the entire decision space is divided into N_T sub-decision spaces $X_1^k, k = 1, \ldots, N_T$ in the first layer.

The proposed region division offers two main advantages. First, the decision space is partitioned in a principled manner

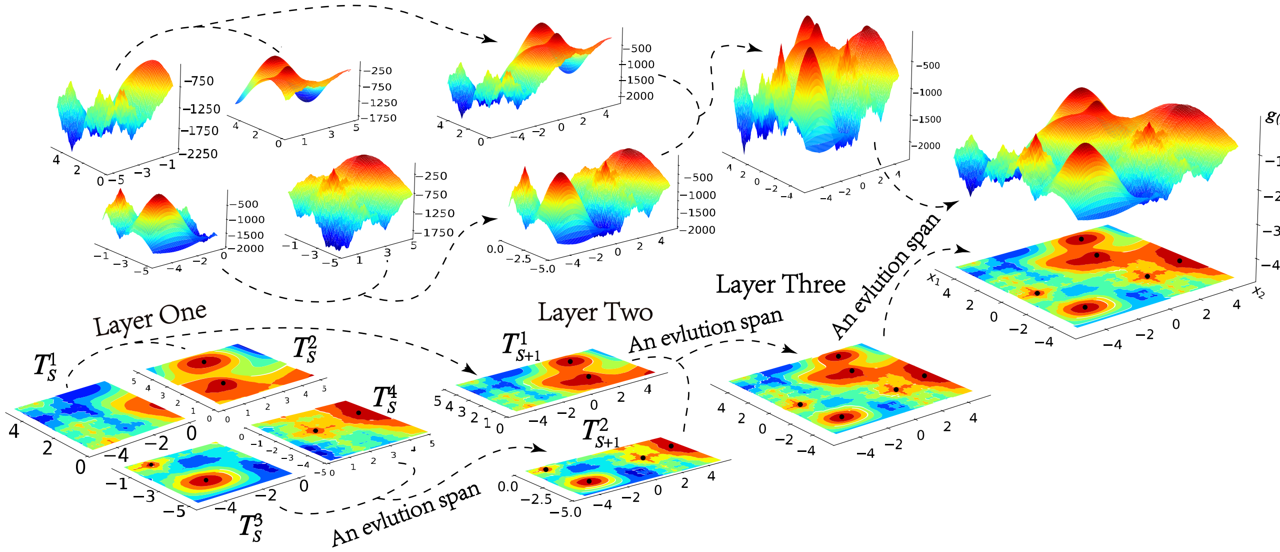


Fig. 2. Illustration of the proposed RDM. The two-dimensional decision space is divided into four sub-decision spaces (T_1^1, T_1^2, T_1^3, and T_1^4) at the initial stage, then gradually merges into two sub-decision spaces (T_2^1 and T_2^2) and finally into the original decision space at the middle and later stages, respectively. Throughout this process, fitness landscapes of varying sizes are shown. Black points indicate multiple optimal solutions.

```
Algorithm 1: Region Division
Input:
  • $P_S$: the sample size;
  • $N_T$: the number of tiles for the first layer.
Division:
  • Use Latin hypercube sampling to generate $P_S$ sample points in the decision space $X$;
  • Archive these $P_S$ sample points as $P = \{\mathbf{r}_1, \ldots, \mathbf{r}_{P_S}\}$;
  • Use the K-means clustering algorithm to group the $P_S$ sample points $\{\mathbf{r}_1, \ldots, \mathbf{r}_{P_S}\}$ into $N_T$ tiles $T_1^k = \{\mathbf{r}_1, \ldots, \mathbf{r}_{p_S^k}\}, k = 1, \ldots, N_T$;
Output: $\{T_1^1, \ldots, T_1^{N_T}\}$ and $P$
```

```
Algorithm 2: Region Merging
Input:
  • $N_T$: the number of tiles in the current layer $s$;
  • $T_s$: the set of $N_T$ tiles $\{T_s^1, \ldots, T_s^{N_T}\}$.
Merging:
Set $T_{s+1}^i = \{\}, i = 1, \ldots, N_T/2$;
for $i$=1 to $N_T/2$ do
    Randomly choose two tiles $T_s^m$ and $T_s^n$ from $T_s$;
    Let $T_{s+1}^i = T_s^m \cup T_s^n$;
    Remove $T_s^m$ and $T_s^n$ from $T_s$;
end
Output: $\{T_{s+1}^1, \ldots, T_{s+1}^{N_T/2}\}$
```

using a clustering algorithm, allowing variable information from sample points to be more easily integrated into evolutionary operations compared to relying solely on objective function values. For instance, when assigning individuals to tiles, the minimal Euclidean distance between a given individual and sample points can be readily computed. Second, by employing sampling and clustering, decision spaces of any dimensionality can be divided into a specified number of sub-decision spaces, making the region division approach flexible and adaptable to high-dimensional spaces.

2) Region Merging: Without *a priori* knowledge, we assume that each sub-decision space is equally important for exploring potential attraction basins. However, some sub-decision spaces do not contain any optimal solutions. If computational resources are continuously allocated to these sub-decision spaces, optimal solutions in other sub-decision spaces may remain undiscovered, leading to resource inefficiency.

To address this issue, an effective approach is to merge certain sub-decision spaces into larger ones, allowing the inclusion of optima within the expanded search area. To balance division and merging, a fixed number of layers (N_L) is defined. Between any two layers, there is an evolutionary span of E_S generations. During each span, every sub-decision space is independently explored and exploited using multiobjective optimization. After every E_S generations, the region merging strategy, presented in **Algorithm 2**, is applied to randomly merge one tile with another.

3) Working Principle: To further illustrate the working principle of RDM, we first assume that the entire decision space X has been divided into N_T tiles in the sth layer, where each tile T_s^i covers a small sub-decision space X_s^i, yielding $X = X_s^1 \bigcup \ldots X_s^i \ldots \bigcup X_s^{N_T}$. The transformed TOP in X_s^k is therefore defined as:

$$\text{minimize } F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x})),$$
$$\text{subeject to } \begin{cases} \mathbf{x} = (x_1, \ldots, x_D) \in X_s^k \\ X_s^k \text{ is the } k\text{th sub-decision} \\ \text{space in the } s\text{th layer.} \end{cases}, \tag{11}$$

where $f_1(\mathbf{x})$, $f_2(\mathbf{x})$, and $f_3(\mathbf{x})$ have the same formulations as in (8).

An example of region division is illustrated in Fig. 2, where a two-dimensional decision space is divided into four sub-decision spaces: T_s^1, T_s^2, T_s^3, and T_s^4. Since each of these sub-decision spaces represents only a small part of the entire decision space, the transformed TOP within each of T_s^1, T_s^2, T_s^3, and T_s^4 is simpler than that in (8), at least in terms of the search space. As shown in Fig. 2, compared to the entire decision space, each sub-decision space has at most two optimal solutions and a simplified fitness landscape, making it easier for multiobjective optimization to locate solutions. In this case, unlike in a standard MOP, the strong regularity typically needed to find multiple optimal solutions is unnecessary within such a small sub-decision space.

Prior to merging, the sub-decision space T_s^1, as shown in Fig. 2, does not contain any optimal solutions. After E_S generations, T_s^1 and T_s^2 merge into the larger tile T_{s+1}^1, similarly, T_s^3 and T_s^4 merge into T_{s+1}^2. As a result, computational resources previously allocated to explore T_s^1 can now be utilized to search for the two optimal solutions within T_{s+1}^1. It is worth noting that, to provide a clearer view of the landscape in Fig. 2, tiles are shown as equal-sized and regularly shaped; however, they are, in fact, irregular, as seen in Fig. 1. Moreover, for ease of understanding, the merging strategy pairs adjacent tiles in Figs. 1 and 2.

D. Complete Algorithm TriDE

```
Algorithm 3: TriDE
Input:
  • $P_S$: the population size and the sample size;
  • $N_L$: the number of layers.
Initialization:
  • Calculate the number of tiles $N_T$, evolution span $E_S$, and
    sub-population size $N_S$;
  • Execute Algorithm 1;
  • Copy the $P_S$ sample points from Algorithm 1 as initial individuals;
  • Evaluate $P_S$ individuals according to (1);
  • Partition $P_S$ individuals into $N_T$ sub-populations $P_1, \dots, P_{N_T}$;
  • Set $t = 0$.
Evolutionary Progress:
while termination is not satisfied do
    $t = t + 1$;
    For each $P_t$, generate its offspring population $Q_t$ with
      SHADE [36];
    Evaluate $Q_t$ according to (1);
    $P = P_1 \bigcup \dots \bigcup P_{N_T} \bigcup Q_1 \dots \bigcup Q_{N_T}$;
    Set $P_1 = P_2 = \dots = P_{N_T} = Q_1 = Q_2 = \dots = Q_{N_T} = \emptyset$;
    for i=1 to $2P_S$ do
        Calculate $f_1(\mathbf{x}_i)$, $f_2(\mathbf{x}_i)$, and $f_3(\mathbf{x}_i)$ according to (8);
        Find the sample point $r_j$ nearest to the $i$th individual;
        Get the corresponding tile $t$ to which $r_j$ belongs;
        Assign the $i$th individual to sub-population $P_t$;
    end
    for i=1 to $N_T$ do
        if the size of $P_i$ exceeds $N_S$ then
            Apply improved non-dominated sorting and
              crowding distance sorting to truncate the size;
        else if the size of $P_i$ is less than $N_S$ then
            Randomly select $N_S - |P_i|$ solutions from other
              sub-populations to meet the size;
        end
    end
    if $t$ equals $E_S$ then
        $t = 0$;
        Execute Algorithm 2;
        $N_T = N_T/2$;
        $N_S = 2N_S$;
    end
end
Output: $P_1 \bigcup \dots \bigcup P_{N_T}$.
```

In this section, the complete TriDE approach is described in detail.

At each generation, TriDE maintains the following information:

1) P_S individuals: $\{\mathbf{x}_j | j = 1, \dots, P_S\}$;
2) N_T tiles in the sth layer: $\{T_s^1, \dots, T_s^{N_T}\}$;
3) N_T sub-populations: $\{P_1, \dots, P_{N_T}\}$;
4) Three objective function values of (8) for each individual: $\{(f_1(\mathbf{x}_j), f_2(\mathbf{x}_j), f_3(\mathbf{x}_j)) | j = 1, \dots, P_S\}$.

The pseudo-code of TriDE is outlined in **Algorithm 3**.

In the initialization phase, **Algorithm 1** is executed to divide the decision space X into N_T sub-decision spaces. The P_S sample points from **Algorithm 1** are directly used as initial individuals in **Algorithm 3**. Subsequently, the values for each sub-population size N_S and evolution span E_S are obtained through the calculations shown in Section IV-B. Generally, each sub-population may not precisely match N_S individuals, so individuals from overloaded sub-populations are randomly selected and moved to under-populated ones to balance the sizes.

During the evolutionary progress, SHADE [36] is performed on each of the N_T sub-populations to generate N_S offspring, resulting in $P_S = N_S \cdot N_T$ new offspring. Next, we combine P_S individuals and P_S offspring to calculate their objective function values according to (8), and re-assign them to N_T sub-populations based on their proximity to the nearest sample points (using minimal Euclidean distance). If a sub-population exceeds N_S individuals, improved non-dominated sorting [6] and crowding distance sorting [37] are applied to truncate the size; if it is below N_S, individuals from other sub-populations are randomly selected to reach the target size. After every E_S generations, **Algorithm 2** is executed to merge N_T tiles into $\frac{N_T}{2}$ larger ones for the next layer.

Generally, TriDE starts with several sub-populations and ends with a single population. Key details include:

- In **Algorithm 1**, P_S sample points are randomly generated and remain fixed during the evolutionary process. These P_S sample points are copied as initial individuals to evolve in subsequent generations. Sample points serve two roles in TriDE: first, they construct the $\alpha(\mathbf{x})$ of (9), which forms the basis of objective conflicts in the proposed transformation (8). Second, these sample points are used to estimate and partition the entire decision space.
- **Algorithm 1** is only executed in the initialization phase. Once N_T initial tiles are obtained, subsequent layers are generated by **Algorithm 2**. An example with $N_T = 8$ is shown in Fig. 1. After each E_S generations, the number of tiles is halved, and in the final E_S generations, only one tile (the original decision space) remains.
- The nearest neighbor classification, shown in Fig. S.2 (supplementary material), is used to allocate individuals to different sub-decision spaces. Based on the Euclidean distance, the sample point closest to a given individual determines its tile assignment, as each sample point belongs to a unique cluster.
- Without *a priori* knowledge, dividing the entire decision space by a definitive formulation is challenging. However,

individuals from different sub-decision spaces evolving in a collaborative manner help maintain population diversity [38]. Consequently, TriDE does not restrict individuals and their offspring to remain solely within their designated sub-decision spaces.

E. Computational Time Complexity

Regarding the computational complexity of TriDE, its primary components in each generation are considered as follows:

1) Calculating the values of $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ and applying the nearest neighbor classification method for each individual $\mathbf{x}$ requires summing the Euclidean distances between $\mathbf{x}$ and P_S sample points. The time complexity for this part is $O(P_S^2)$.
2) Calculating the $f_3(\mathbf{x})$ value for each individual requires the Euclidean distance between $\mathbf{x}$ and every other individual. This also has a time complexity of $O(P_S^2)$.
3) For non-dominated sorting, the worst-case time complexity is $O(N_T \cdot N_S^2)$.
4) For crowding distance sorting, the worst-case time complexity is $O(N_T \cdot N_S \cdot \log(N_S))$.

In this study, $P_S = N_T \cdot N_S$, implying that $O(P_S^2) > O(N_T \cdot N_S^2)$. Additionally, the K-means clustering algorithm, applied during the initialization phase, has a time complexity of $O(P_S^2)$. Considering all components, the overall time complexity of TriDE can be simplified to $O(P_S^2)$.

IV. Experimental Study

A. Test Suite and Performance Metrics

In this section, the performance of locating multiple optimal solutions is evaluated on the test suit used in the IEEE WCCI/CEC 2020 competition for multimodal optimization [39]. This suite consists of 20 benchmark functions, categorized into three groups. The first group includes 10 widely used benchmark functions (F_1 to F_{10})); the second group comprises five low-dimensional composition test functions (F_{11} to F_{15}); and the third group contains five high-dimensional composition test functions (F_{16} to F_{20}). The maximum number of fitness evaluations (MaxFEs) and characteristics of these 20 functions are presented in Table S.I and Fig. S.3 (supplementary material).

The performance of the proposed TriDE algorithm is compared with 14 state-of-the-art algorithms based on two evaluation criteria: peak ratio (PR) and success rate (SR). For a specified accuracy level ϵ, PR represents the average ratio of acceptable global optima identified by the algorithm across all trials, while SR indicates the proportion of successful trials to total trials. In this paper, PR and SR values are calculated at five accuracy levels: $\epsilon = 1.0 \times 10^{-1}, 1.0 \times 10^{-2}, 1.0 \times 10^{-3}, 1.0 \times 10^{-4}$, and 1.0×10^{-5}. The methods for calculating PR and SR are detailed in Appendix B (supplementary material).

B. Parameter Settings

TriDE introduces only two parameters: the population size P_S and the number of layers N_L. In this study, P_S is set to 640, and N_L is set to $\max\{2, \lfloor \frac{MaxFEs}{1.0\times 10^5} \rfloor\}$, adjusted according to $MaxFEs$. The minimum value for N_L is set to 2, with larger values for N_L allowed as the number of fitness evaluations increases. The way to set niche radius σ is given in Appendix C. The evolutionary span E_S between layers and the number of tiles N_T for the first layer are automatically set to $E_S = MaxFEs/(P_S \cdot N_L)$ and $N_T = 2^{N_L}$, respectively. TriDE is evaluated using 25 independent trials for each test function.

C. Comparison with State-of-the-Art Multimodal Optimization Algorithms

The performance of the proposed TriDE algorithm is compared with the 14 state-of-the-art algorithms, categorized as follows:

- Recent multimodal optimization algorithms published in prominent journals within the past five years, including FBK-DE [40], NBNC-PSO-ES [41], PMODE [42], ESPDE [43], NDC-DE [44], DIDE [45], ANDE [46], NetCDE$_{\text{MMOPs}}$ [47], and AED-DDE [20].
- Winners of recent IEEE CEC competitions on multimodal optimization, including NEA2 [48], NMMSO [32], RS-CMSA-ES [49], and RS-CMSA-ESII [50], with RS-CMSA-ESII being the winner of the latest competition.
- A recent evolutionary algorithm based on multiobjective optimization, namely EMO-MMO [7], specifically developed for multimodal optimization.

Parameter settings for these algorithms follow the recommendations provided in their respective references.

The PR and SR values obtained by TriDE across five accuracy levels are reported in Table S.II (supplementary material). Since the accuracy level $\varepsilon = 1.0 \times 10^{-4}$ is commonly used for the discussions of comparisons and analyses [46], PR and SR values at this accuracy level are compared with those of other state-of-the-art algorithms in Table S.III (supplementary material). In Table S.III, the best PR values for each test function are highlighted in **boldface**. Symbols $+$, $\approx$, and $-$ indicate that TriDE's PR value is better than, equal to, or worse than that of the compared algorithm, respectively.

Table S.III shows that TriDE attains the best performance on 14 test functions, consistently identifying all global peaks (SR=1.000) on 13 functions at the accuracy of $\varepsilon = 1.0 \times 10^{-4}$. These two numbers are both the best compared with those of other state-of-the-art algorithms. Specifically, TriDE achieves PR and SR values of 1.000 for functions F_1 to F_{10}, indicating that TriDE reliably finds all optimal solutions in this first group of test problems. Among the five low-dimensional composition functions (F_{11} to F_{15}) in the second group, TriDE perfectly solves problems F_{11} to F_{13}, while maintaining robust performance on F_{14} and F_{15}. Additionally, based on PR and SR values at other accuracy levels (see Table S.II), TriDE demonstrates notable stability, as PR and SR values remain largely unaffected even at stricter accuracy ($\varepsilon = 1.0 \times 10^{-5}$).

Furthermore, TriDE outperforms eight compared algorithms on at least six test functions, and two compared algorithms cannot outperform TriDE on any function. Compared to the 13 single-objective-based algorithms, TriDE exhibits marked advantages, particularly on functions F_7, F_8, and F_9, which feature numerous global optima. Despite the inclusion of various niching methods and diversity enhancement mechanisms, none of these comparative algorithms consistently identify all global optima across 25 consecutive trials. Compared to the bi-objective optimization-based algorithm EMO-MMO, TriDE demonstrates substantial improvement on the last ten test functions, outperforming EMO-MMO on seven of them.

Fig. S.3 presents the fitness landscape and the candidate solutions obtained by TriDE on ten selected test functions. The fitness landscapes of these functions, displayed in two- or three-dimensional diagrams, reveal insights into the different attraction basins for local and global optima. As shown in Fig. S.3, despite complex fitness landscapes with numerous peaks and uneven distributions of global optima, TriDE effectively locates all optimal solutions at high accuracy levels, indicating a balanced capability for exploration and exploitation on these functions.

In summary, the comparative analyses and statistical evaluations clearly demonstrate the robust and consistent performance of the proposed TriDE algorithm in identifying multiple optimal solutions across the benchmark test suite.

D. Component Analysis

Since the performance of TriDE is highly dependent on the RDM strategy and tri-objective transformation, empirical experiments were conducted to evaluate their influences.

1) Influence of Tri-objective Transformation: Unlike the commonly used bi-objective transformation for locating multiple optima, TriDE incorporates an additional density function $f_3(\mathbf{x})$ to further enhance population diversity. To assess the effectiveness of $f_3(\mathbf{x})$, we developed a variant, denoted as TriDE-1, which excludes $f_3(\mathbf{x})$ from (11), thus relying solely on the two objective functions $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ in combination with the RDM strategy. Experimental results for TriDE-1 are presented in Table S.IV (supplementary material).

For the first 10 test functions (F_1 to F_{10}), both TriDE and TriDE-1 consistently found all optimal solutions across 25 independent trials. However, for the remaining 10 test functions (F_{11} to F_{20}), the performance of TriDE-1 deteriorated significantly. Overall, TriDE did not lose to TriDE-1 on any test function. Two points can be drawn from this comparison. First, while $f_3(\mathbf{x})$ does not conflict with $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$, it does not compromise the Pareto dominance introduced by the conflicting objectives $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$. Second, the density function $f_3(\mathbf{x})$ aids the evolutionary search by improving the distinction between candidate solutions. As noted in [45], composite test functions have complex fitness landscapes and a greater number of local optima. Therefore, with only $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$, the transformed global and local optima could have narrow regions, as depicted in Fig. S.1(b). The inclusion of the density function increases the distance between candidate solutions. Based on the comparison between TriDE and TriDE-1, this added separation proves beneficial for evolutionary search in distinguishing different individuals during the optimization process.

In conclusion, the density function is highly effective in the proposed transformation for handling multimodality.

2) Influence of RDM Strategy: To evaluate the effectiveness of the RDM strategy, the transformation (8) rather than (11) is applied directly to the entire decision space in this subsection. This variant is denoted as TriDE-2.

Empirical results for TriDE-2 are summarized in Table S.V (supplementary material). The performance of TriDE-2 significantly declines across most test functions, including two relatively simple functions, F_4 and F_9. In contrast, TriDE demonstrates superior competence over TriDE-2 on the test suite, especially in handling the last 10 composition test functions. Additionally, PR values obtained by TriDE-2 are 0.220, 0.167, 0.125, and 0.125 on functions F_{17} to F_{20}, respectively, indicating that TriDE-2 fails to consistently locate multiple optima and ultimately converges toward a single optimal solution in each trial.

From these comparisons, it can be inferred that TriDE-2 is ineffective in seeking multiple optima on high-dimensional test functions. This poor performance is likely due to an inability to explore the uneven distribution of attraction basins within the decision space. In contrast, TriDE achieves a higher accuracy level in locating multiple optimal solutions, underscoring the effectiveness of the RDM strategy in balancing exploration and exploitation.

3) Influence of the Way to Merge Two Tiles: In practice, unlike the illustration in Fig. 2, the boundaries of the divided tiles are often irregular, and the sequence of merging tiles is complex, especially in high-dimensional decision spaces. An example is given in Fig. S.2. When the tile estimated by blue points merges with any one of its adjacent tiles, the other two tiles remain nonadjacent. In TriDE, the merging operation is simplified by randomly selecting two tiles to merge. To investigate the impact of this simplification, two alternative merging operations are designed. The first utilizes the K-means clustering algorithm directly. When merging is activated, $\frac{N_T}{2}$ tiles for the next layer are generated by K-means clustering instead of unifying two tiles. The second approach uses the centroid of each tile to find the nearest neighboring tile for merging. Details can be found in **Algorithm S.1** (supplementary material). Accordingly, two variants, namely TriDE-KM and TriDE-NN, were developed for the first and second merging operations, respectively.

Experimental results obtained by TriDE, TriDE-KM, and TriDE-NN are presented in Table S.VI (supplementary material). The results show that the different merging operations achieve nearly the same performance. For the first 13 test functions, the performance is identical, and for the last 7 high-dimensional test functions, the K-means clustering-based merging operation slightly outperforms the others. As the number of variables increases, the spatial relationships between tiles become more complex. For example, one tile may be adjacent to multiple tiles, as illustrated by the blue domain in Fig. S.2. When TriDE and TriDE-NN are applied to the four tiles in Fig. S.2, no matter which two tiles are selected to merge, the remaining two will have a separation after merging. In contrast, TriDE-KM can reasonably generate $\frac{N_T}{2}$ new tiles using K-means clustering, resulting in better-partitioned subdecision spaces that benefit evolutionary search. However, the time complexity of the K-means clustering algorithm is $O(P_S^2)$, meaning that TriDE-KM requires more execution time than TriDE.

Considering both performance and execution time, TriDE is adopted in this paper.

E. Parameter Analysis

1) Sensitivity of the Population Size: The sensitivity of population size (P_S) has been investigated in this section. Three variants, namely TriDE-3, TriDE-4, and TriDE-5, were developed with P_S values set to 384, 480, and 768, respectively. Table S.VII (supplementary material) summarizes the empirical results of PR and SR values obtained by these variants.

For TriDE-3 and TriDE-4, P_S is smaller than that of TriDE, allowing for a comparatively larger number of generations to evolve. In contrast, TriDE-5 initiates a larger population size, resulting in a shorter evolutionary process. Table S.VII shows that TriDE and its three variants (TriDE-3, TriDE-4, and TriDE-5) achieve similar performance on the first 15 test functions (F_1 to F_{15}), while the performance of TriDE-3, TriDE-4, and TriDE-5 is noticeably lower than that of TriDE on the last five high-dimensional test functions (F_{16} to F_{20}).

During the initialization phase, P_S individuals are equally distributed across N_T tiles, so each tile contains $\frac{P_S}{N_T}$ individuals. If $\frac{P_S}{N_T}$ is too small, the sub-population lacks sufficient individuals to adequately explore the corresponding sub-decision space at the initial stage, potentially causing some attraction basins to be overlooked. Conversely, if P_S is set too high, the generation count ($\frac{MaxFEs}{P_S}$) becomes low, leaving the sub-population with too few generations to fully exploit promising solutions within the attraction basins found. Accordingly, the poor performance of TriDE-3 and TriDE-4 can be attributed to insufficient individuals for exploring each tile in the initial and middle stages, while TriDE-5 lacks sufficient evolutionary generations to exploit attraction basins in the later stage.

Based on the above discussion, it is observed that $P_S = 640$ achieves a balanced trade-off between exploration and exploitation across the 20 multimodal test functions, and is thus recommended for TriDE.

2) Sensitivity of the Number of Sample Points: In our proposed approach, sample points are used to construct the conflicting objectives $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ and to estimate the decision space for division. A larger number of sample points can better estimate the decision space, resulting in improved division from the K-means clustering algorithm. However, if the number of sample points is too large, the decision space becomes densely populated with sample points, causing them to lose their role as effective reference markers for constructing conflicting objectives. To ascertain the sensitivity of sample size, four variants, namely TriDE-6, TriDE-7, TriDE-8, and TriDE-9, were developed with 400, 600, 700, and 800 sample points, respectively.

According to the empirical results presented in Table S.VIII (supplementary material), the number of sample points mainly impacts performance on high-dimensional composition test functions F_{16} to F_{20}. For low-dimensional composition test functions F_{11} to F_{15}, the performance difference between TriDE and its variants (TriDE-6 to TriDE-9) is minimal. For instance, TriDE-8 with 700 sample points achieves PR values of 0.98, 0.860, and 0.730 on F_{13}, F_{14}, and F_{15}, respectively, which are similar to the performance of TriDE. For high-dimensional composition test functions F_{16} to F_{20}, TriDE, TriDE-6, and TriDE-7, with the number of sample points is set to 500, 400, and 600, respectively, achieve the best PR values on three test functions. In contrast, TriDE-8 and TriDE-9, with 700 and 1000 sample points, respectively, exhibit slightly lower performance.

Based on these comparisons, a moderate number of sample points–around 500–is more suitable for TriDE in constructing conflicting objectives and effectively sampling the decision space for division.

F. Application of TriDE to Real-world MMOPs

To validate the practical applicability of the proposed TriDE algorithm, four real-world MMOPs–the multiple steady states problem (P1), molecular conformation problem (P2), robot kinematic problem (P3), and interval arithmetic problem (P4)–are selected from nonlinear equation systems [51]. For comparative evaluation, two recent multimodal optimization algorithms, DIDE and ANDE, published in reputable journals within the past five years, along with HillVallEA19 [52] and RS-CMSA-ESII, the winners of the most recent IEEE CEC and GECCO competitions, respectively, are chosen. The maximum number of function evaluations (MaxFEs) and accuracy level parameters are configured according to [51]. Detailed comparison results in terms of peak ratio (PR) between TriDE and these four algorithms for the selected real-world MMOPs are presented in Table S.IX (supplementary material). Symbols $+$, $\approx$, and $-$ respectively indicate whether the PR values obtained by TriDE are better, similar, or worse compared to those of the referenced algorithms.

As indicated in Table S.IX, TriDE outperforms all compared multimodal optimization algorithms on three out of the four selected real-world MMOPs. Specifically, on problem P1, only RS-CMSA-ESII achieves performance comparable to TriDE, while TriDE significantly outperforms the other algorithms. For problem P3, both RS-CMSA-ESII and TriDE demonstrate excellent performance, with PR and SR values equal to 1.000, surpassing DIDE, ANDE, and HillVallEA19. Regarding problem P4, nearly all evaluated algorithms achieve satisfactory results with PR and SR values of 1.000.

In summary, the proposed TriDE algorithm demonstrates superior performance over four state-of-the-art algorithms on selected real-world MMOPs, affirming its strong practical applicability.

V. Conclusion

In this study, a novel RDM-based multiobjective evolutionary approach, termed TriDE, is proposed to efficiently locate multiple optimal solutions. TriDE transforms a given MMOP into a TOP, subsequently solving the transformed TOP through multiobjective optimization. In this framework, the multiple global optima of the original MMOP are identified as the Pareto-optimal solutions of the transformed TOP. Since the multiple optimal solutions are discretely distributed, an RDM strategy is introduced to balance exploration and exploitation throughout the evolutionary process. Under the region division strategy, the entire decision space is partitioned into numerous tiles to promote exploration within each sub-decision space. In the merging strategy, additional computational resources are allocated to exploit discovered attraction basins, thereby achieving candidate solutions with high accuracy.

The performance of the proposed TriDE algorithm is evaluated on 20 benchmark test functions and compared against 14 state-of-the-art multimodal optimization algorithms. TriDE achieves optimal performance on 13 out of the 20 test functions and demonstrates significant improvements over the compared algorithms on most functions. Additional analyses validate the effectiveness of the proposed tri-objective transformation and the RDM strategy in identifying multiple optimal solutions. Furthermore, the applicability of TriDE is illustrated through experiments conducted on selected real-world MMOPs.

Future research will focus on improving the performance of TriDE on more complicated MMOPs and further expanding the application of TriDE to more complex real-world scenarios.

REFERENCES

[1] M. Sarkar, J. Pradhan, A. Kumar Singh, and H. Nenavath, "A novel hybrid quantum architecture for path planning in quantum-enabled autonomous mobile robots," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 3, pp. 5597–5606, 2024.

[2] Y.-J. Gong, T. Huang, Y.-N. Ma, S.-W. Jeon, and J. Zhang, "Mtrajplanner: A multiple-trajectory planning algorithm for autonomous underwater vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 4, pp. 3714–3727, 2023.

[3] J. Wu, H. Yang, Y. Zeng, Z. Wu, J. Liu, and L. Feng, "A twin learning framework for traveling salesman problem based on autoencoder, graph filter, and transfer learning," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 3245–3258, 2024.

[4] S.-J. Jian and S.-Y. Hsieh, "A niching regression adaptive memetic algorithm for multimodal optimization of the euclidean traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 5, pp. 1413–1426, 2023.

[5] S. Cheng, H. Jin, H. Lu, and Y. Shi, "A Q-learning based brainstorming optimization algorithm for solving multimodal optimization problems," *IEEE Transactions on Consumer Electronics*, pp. 1–1, 2024.

[6] Y. Wang, H.-X. Li, G. G. Yen, and W. Song, "MOMMOP: Multiobjective optimization for locating multiple optimal solutions of multimodal optimization problems," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 830–843, 2015.

[7] R. Cheng, M. Li, K. Li, and X. Yao, "Evolutionary multiobjective optimization-based multimodal optimization: Fitness landscape approximation and peak detection," *IEEE Trans. Evol. Comput.*, vol. 22, no. 5, pp. 692–706, 2018.

[8] J. Feng, B. Huang, X. Zhou, and X. Jin, "Large-scale multi-agent learning-based cloud-edge collaborative distributed PV data compression and information aggregation for multimodal network in power systems," *IEEE Transactions on Consumer Electronics*, pp. 1–1, 2024.

[9] Y.-G. Wang, L. Fan, and Y.-Q. Lei, "PSO-based robust watermarking of AVS-encoded video," in *2010 IEEE International Conference on Multimedia and Expo*, 2010, pp. 1647–1650.

[10] A. P. Muniyandi, B. Balusamy, R. K. Dhanaraj, D. Sumathi, S. Nandakumar, K. S. Preetha, R. Alroobaea, and A. Paramasivam, "Intelligent security system for preventing DDoS attacks for 6G enabled WBSN using improve grey wolf optimization," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 3, pp. 5775–5782, 2024.

[11] Y.-G. Wang, Z.-M. Lu, L. Fan, and Y. Zheng, "Robust dual watermarking algorithm for AVS video," *Signal Processing: Image Communication*, vol. 24, no. 4, pp. 333–344, 2009, special Issue on AVS and its Application.

[12] Y. Kang, L. Peng, J. Guo, Y. Lu, Y. Yang, B. Fan, and B. Pu, "A fast hybrid feature selection method based on dynamic clustering and improved particle swarm optimization for high-dimensional health care data," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 2447–2459, 2024.

[13] X. Li, M. G. Epitropakis, K. Deb, and A. Engelbrecht, "Seeking multiple solutions: An updated survey on niching methods and their applications," *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 518–538, 2017.

[14] B. Sareni and L. Krahenbuhl, "Fitness sharing and niching methods revisited," *IEEE Trans. Evol. Comput.*, vol. 2, no. 3, pp. 97–106, 1998.

[15] X. Li, "Niching without niching parameters: Particle swarm optimization using a ring topology," *IEEE Trans. Evol. Comput.*, vol. 14, no. 1, pp. 150–169, 2010.

[16] A. Petrowski, "A clearing procedure as a niching method for genetic algorithms," in *Proceedings of IEEE International Conference on Evolutionary Computation*, 1996, pp. 798–803.

[17] R. Thomsen, "Multimodal optimization using crowding-based differential evolution," in *Proceedings of the 2004 Congress on Evolutionary Computation*, vol. 2, 2004, pp. 1382–1389 Vol.2.

[18] S. Biswas, S. Kundu, and S. Das, "Inducing niching behavior in differential evolution through local information sharing," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 246–263, 2015.

[19] W. Gao, G. G. Yen, and S. Liu, "A cluster-based differential evolution with self-adaptive strategy for multimodal optimization," *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1314–1327, 2014.

[20] Z.-J. Wang, Y.-R. Zhou, and J. Zhang, "Adaptive estimation distribution distributed differential evolution for multimodal optimization problems," *IEEE Trans. Cybern.*, pp. 1–12, 2020.

[21] Q. Yang, W.-N. Chen, Y. Li, C. L. P. Chen, X.-M. Xu, and J. Zhang, "Multimodal estimation of distribution algorithms," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 636–650, 2017.

[22] L. Li and K. Tang, "History-based topological speciation for multimodal optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 1, pp. 136–150, 2015.

[23] Z.-J. Wang, Z.-H. Zhan, Y. Lin, W.-J. Yu, H.-Q. Yuan, T.-L. Gu, S. Kwong, and J. Zhang, "Dual-strategy differential evolution with affinity propagation clustering for multimodal optimization problems," *IEEE Trans. Evol. Comput.*, vol. 22, no. 6, pp. 894–908, 2018.

[24] S. Biswas, S. Kundu, and S. Das, "An improved parent-centric mutation with normalized neighborhoods for inducing niching behavior in differential evolution," *IEEE Trans. Cybern.*, vol. 44, no. 10, pp. 1726–1737, 2014.

[25] B. Y. Qu, P. N. Suganthan, and S. Das, "A distance-based locally informed particle swarm model for multimodal optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 3, pp. 387–402, 2013.

[26] J. Wang, C. Li, S. Zeng, and S. Yang, "History-guided hill exploration for evolutionary computation," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 6, pp. 1962–1975, 2023.

[27] B. Y. Qu, P. N. Suganthan, and J. J. Liang, "Differential evolution with neighborhood mutation for multimodal optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 5, pp. 601–614, 2012.

[28] K. Deb and A. Saha, "Multimodal optimization using a bi-objective evolutionary algorithm," *Evol Comput*, vol. 20, no. 1, pp. 27–62, 03 2012.

[29] J. Yao, N. Kharma, and P. Grogono, "Bi-objective multipopulation genetic algorithm for multimodal function optimization," *IEEE Trans. Evol. Comput.*, vol. 14, no. 1, pp. 80–102, 2010.

[30] W. Song, Y. Wang, H.-X. Li, and Z. Cai, "Locating multiple optimal solutions of nonlinear equation systems based on multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 3, pp. 414–431, 2015.

[31] A. Basak, S. Das, and K. C. Tan, "Multimodal optimization using a biobjective differential evolution algorithm enhanced with mean distance-based selection," *IEEE Trans. Evol. Comput.*, vol. 17, no. 5, pp. 666–685, 2013.

[32] J. E. Fieldsend, "Running up those hills: Multi-modal search with the niching migratory multi-swarm optimiser," in *Proc. IEEE Congr. Evol. Comput.*, 2014, pp. 2593–2600.

[33] Q. Fan and X. Yan, "Solving multimodal multiobjective problems through zoning search," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 8, pp. 4836–4847, 2021.

[34] M. D. Mckay, R. J. Beckman, and W. J. C. and, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 42, no. 1, pp. 55–61, 2000.

[35] D. Wang, J. Lin, and Y.-G. Wang, "Query-efficient adversarial attack based on Latin hypercube sampling," in *2022 IEEE International Conference on Image Processing (ICIP)*, 2022, pp. 546–550.

[36] R. Tanabe and A. Fukunaga, “Success-history based parameter adaptation for differential evolution,” in *Proc. IEEE Congr. Evol. Comput.*, 2013, pp. 71–78.

[37] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.

[38] H.-L. Liu, F. Gu, and Q. Zhang, “Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems,” *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 450–455, 2014.

[39] M. G. Epitropakis, M. Preuss, and X. Li, “Results of the WCCI/CEC 2020 competition on niching methods for multimodal optimization.” [Online]. Available: https://github.com/mikeagn/CEC2013/blob/master/Niching-final-presentation-CEC2020.pdf

[40] X. Lin, W. Luo, and P. Xu, “Differential evolution for multimodal optimization with species by nearest-better clustering,” *IEEE Transactions on Cybernetics*, vol. 51, no. 2, pp. 970–983, 2021.

[41] W. Luo, Y. Qiao, X. Lin, P. Xu, and M. Preuss, “Hybridizing niching, particle swarm optimization, and evolution strategy for multimodal optimization,” *IEEE Transactions on Cybernetics*, vol. 52, no. 7, pp. 6707–6720, 2022.

[42] Z. Wei, W. Gao, G. Li, and Q. Zhang, “A penalty-based differential evolution for multimodal optimization,” *IEEE Transactions on Cybernetics*, vol. 52, no. 7, pp. 6024–6033, 2022.

[43] Y. Li, L. Huang, W. Gao, Z. Wei, T. Huang, J. Xu, and M. Gong, “History information-based hill-valley technique for multimodal optimization problems,” *Information Sciences*, vol. 631, pp. 15–30, 2023.

[44] Y. Sun, G. Pan, Y. Li, and Y. Yang, “Differential evolution with nearest density clustering for multimodal optimization problems,” *Information Sciences*, vol. 637, p. 118957, 2023.

[45] Z.-G. Chen, Z.-H. Zhan, H. Wang, and J. Zhang, “Distributed individuals for multiple peaks: A novel differential evolution for multimodal optimization problems,” *IEEE Trans. Evol. Comput.*, vol. 24, no. 4, pp. 708–719, 2020.

[46] Z.-J. Wang, Z.-H. Zhan, Y. Lin, W.-J. Yu, H. Wang, S. Kwong, and J. Zhang, “Automatic niching differential evolution with contour prediction approach for multimodal optimization problems,” *IEEE Trans. Evol. Comput.*, vol. 24, no. 1, pp. 114–128, 2020.

[47] X.-Y. Chen, H. Zhao, and J. Liu, “A network community-based differential evolution for multimodal optimization problems,” *Information Sciences*, vol. 645, p. 119359, 2023.

[48] M. Preuss, “Niching the CMA-ES via nearest-better clustering,” in *Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation*, 2010, pp. 1711–1718.

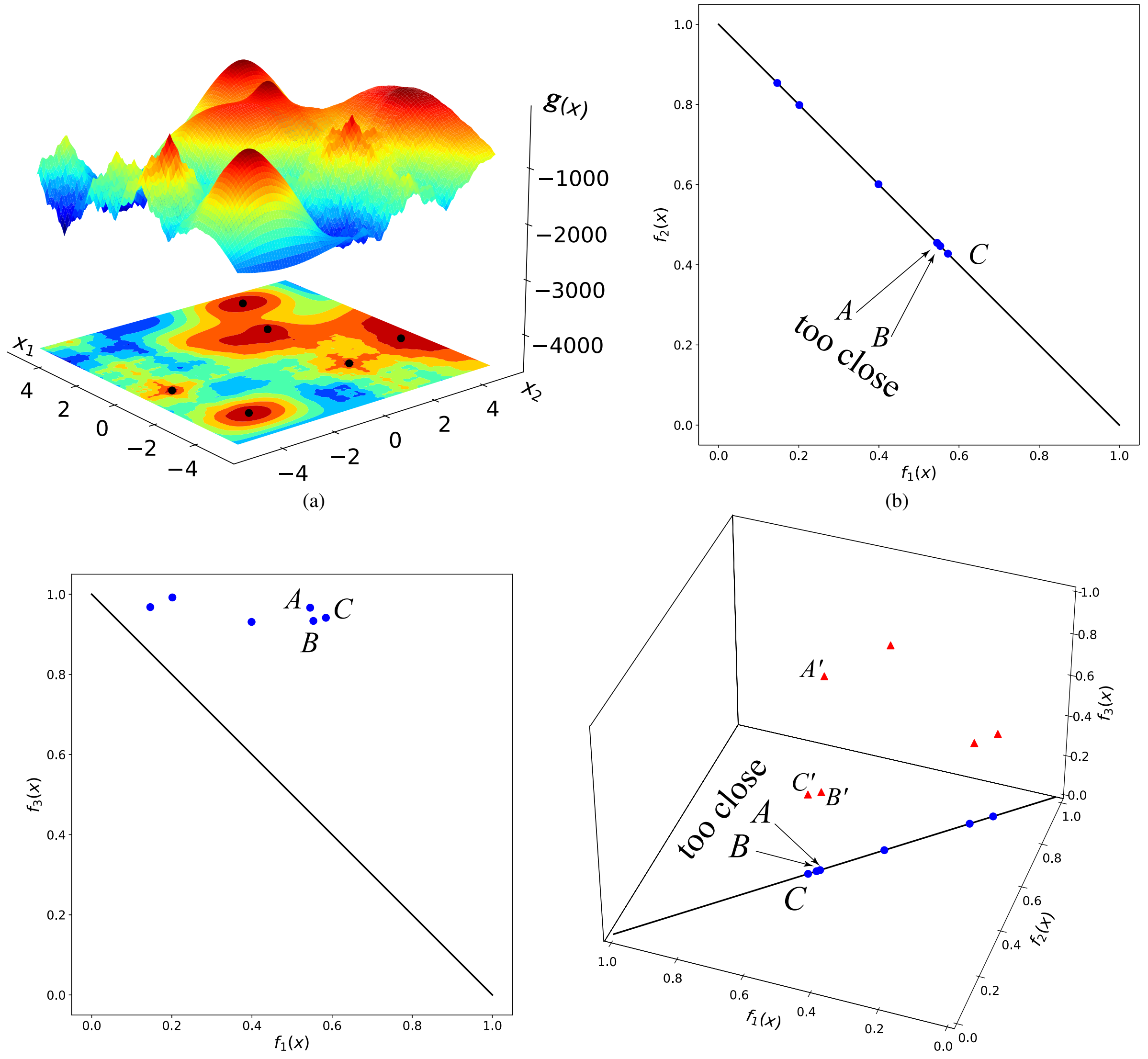
[49] A. Ahrari, K. Deb, and M. Preuss, “Multimodal optimization by covariance matrix self-adaptation evolution strategy with repelling subpopulations,” *Evolutionary Computation*, vol. 25, no. 3, pp. 439–471, 2017.

[50] A. Ahrari, S. Elsayed, R. Sarker, D. Essam, and C. A. C. Coello, “Static and dynamic multimodal optimization by improved covariance matrix self-adaptation evolution strategy with repelling subpopulations,” *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 3, pp. 527–541, 2022.

[51] W. Gong, Y. Wang, Z. Cai, and L. Wang, “Finding multiple roots of nonlinear equation systems via a repulsion-based adaptive differential evolution,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 4, pp. 1499–1513, 2020.

[52] S. Maree, T. Alderliesten, and P. A. Bosman, “Benchmarking hillvallea for the GECCO 2019 competition on multimodal optimization,” *arXiv preprint arXiv:1907.10988*, 2019.

Supplementary Material for “Region Division and Merging-based Multiobjective Optimization for Multimodal Problems”



(a)

(b)

(c)

(d)

Fig. S.1. Schematic illustration of the proposed tri-objective transformation. Black points indicate six optimal solutions. Blue and red points represent candidate solutions in the bi-objective and tri-objective transformations, respectively. (a) The fitness landscape of a specific MMOP. (b) Bi-objective transformation under $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$. (c) Bi-objective transformation under $f_1(\mathbf{x})$ and $f_3(\mathbf{x})$. (d) Tri-objective transformation.

Algorithm S.1: Nearest Neighbor Tile Merging

```
Input:
  • $N_T$: the number of tiles in the current layer $s$;
  • $C_s$: the set of centroid of $N_T$ tiles $\{\mathbf{c}_s^1, \ldots, \mathbf{c}_s^{N_T}\}$ ;      /* The initial centroid of each tile can be obtained by
    K-means clustering algorithm in the initialization phase. */
  • $T_s$: the set of $N_T$ tiles $\{T_s^1, \ldots, T_s^{N_T}\}$.
Merging:
Set $T_{s+1}^i = \{\}, i = 1, \ldots, N_T/2$;
while $T_s$ is not empty do
    Set $D_S = \{\}$;
    for $i$=1 to $|T_s|$ do
        for $j$=$i$+1 to $|T_s|$ do
            Calculate the Euclidean distance $D_{ij}$ between $\mathbf{c}_i$ and $\mathbf{c}_j$;
            Put $D_{ij}$ into $D_S$;
        end
    end
    Find the minimum value from $D_S$ and its corresponding tiles $m$ and $n$;
    Let $T_{s+1}^i = T_s^m \cup T_s^n$;
    Let $\mathbf{c}_{s+1}^i = \frac{\mathbf{c}_s^m + \mathbf{c}_s^n}{2}$;
    Remove $T_s^m$ and $T_s^n$ from $T_s$;
end
Output: $\{T_{s+1}^1, \ldots, T_{s+1}^{N_T/2}\}$, $\{\mathbf{c}_{s+1}^i, \ldots, \mathbf{c}_{s+1}^{N_T/2}\}$
```

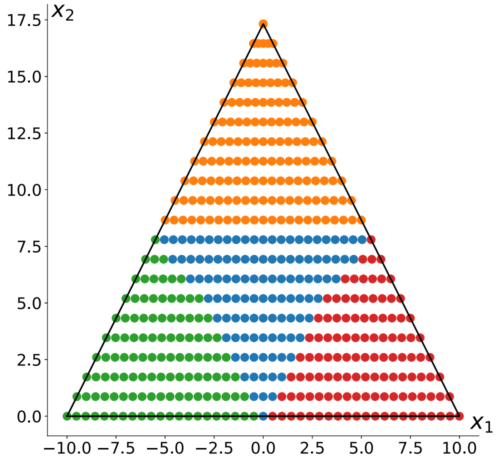
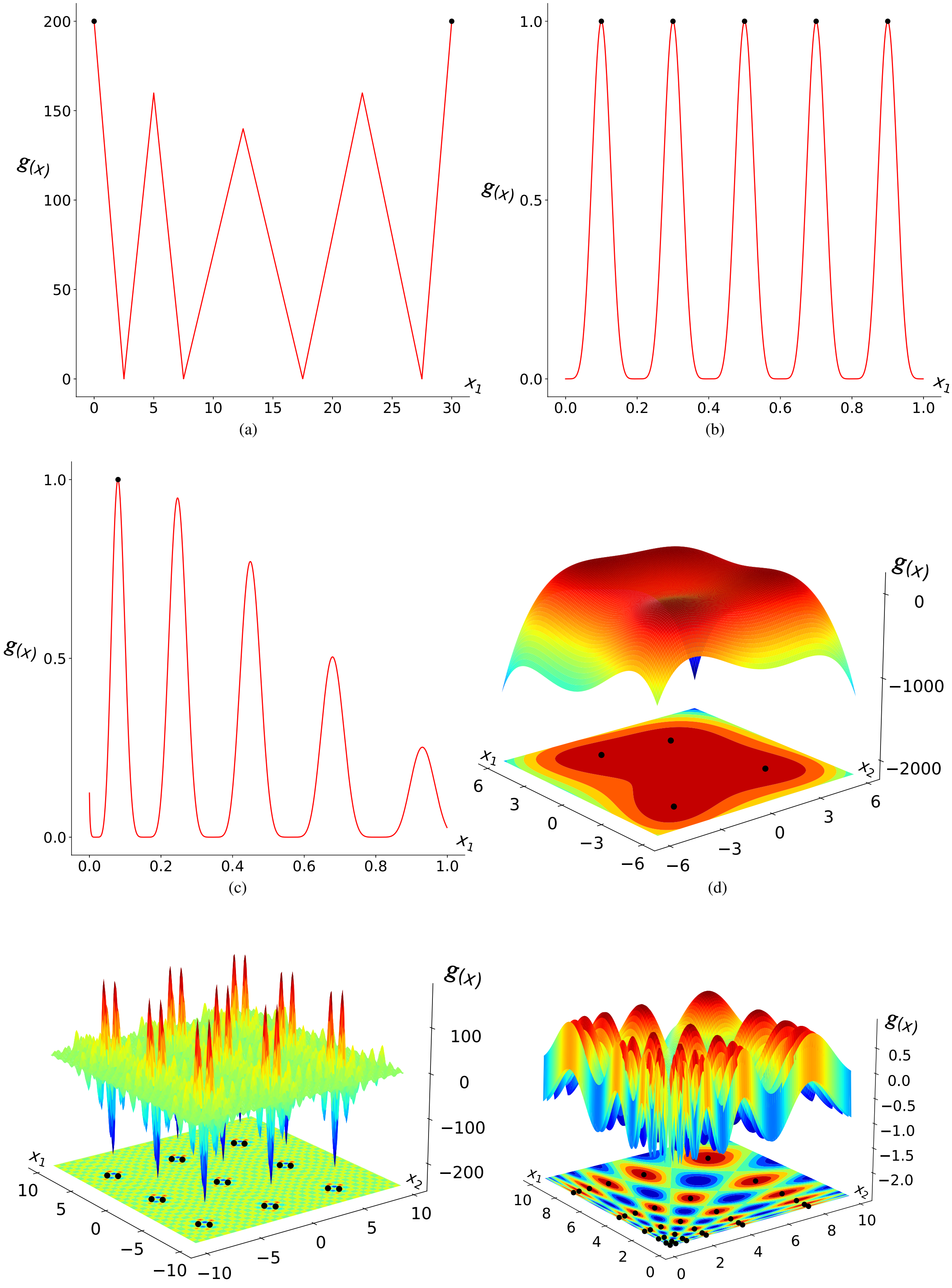


Fig. S.2. Illustration of the merging strategy operation. The triangular domain is divided into four tiles, estimated by orange, blue, green, and red points. The tile estimated by blue points is adjacent to the other three tiles, while any two of the three tiles estimated by orange, green, and red points are nonadjacent.

TABLE S.I. Basic Properties of Benchmark Functions in the IEEE CEC'2013 Test Suite

Function	D	Number of global optima	Number of local optima	r	MaxFEs
F_1: Five-Uneven-Peak Trap	1	2	3	0.01	5.0E+04
F_2: Equal Maxima	1	5	0	0.01	5.0E+04
F_3: Uneven Decreasing Maxima	1	1	4	0.01	5.0E+04
F_4: Himmelblau	2	4	0	0.01	5.0E+04
F_5: Six-Hump Camel Back	2	2	2	0.5	5.0E+04
F_6: Shubert	2	18	many	0.5	2.0E+05
F_7: Vincent	2	36	0	0.2	2.0E+05
F_8: Shubert	3	81	many	0.5	4.0E+05
F_9: Vincent	3	216	0	0.2	4.0E+05
F_{10}: Modified Rastrigin	2	12	0	0.01	2.0E+05
F_{11}: Composition Function 1	2	6	many	0.01	2.0E+05
F_{12}: Composition Function 2	2	8	many	0.01	2.0E+05
F_{13}: Composition Function 3	2	6	many	0.01	2.0E+05
F_{14}: Composition Function 3	3	6	many	0.01	4.0E+05
F_{15}: Composition Function 4	3	8	many	0.01	4.0E+05
F_{16}: Composition Function 3	5	6	many	0.01	4.0E+05
F_{17}: Composition Function 4	5	8	many	0.01	4.0E+05
F_{18}: Composition Function 3	10	6	many	0.01	4.0E+05
F_{19}: Composition Function 4	10	8	many	0.01	4.0E+05
F_{20}: Composition Function 4	20	8	many	0.01	4.0E+05


$g_{(x)}$
200
150
100
50
0
0
5
10
15
20
25
30
x_1
$g_{(x)}$
1.0
0.5
0.0
0.0
0.2
0.4
0.6
0.8
1.0
x_1
$g_{(x)}$
1.0
0.5
0.0
0.0
0.2
0.4
0.6
0.8
1.0
x_1
$g_{(x)}$
0
−1000
−2000
x_1
x_2
6
3
0
−3
−6
−6
−3
0
3
6
$g_{(x)}$
100
0
−100
−200
x_1
x_2
10
5
0
−5
−10
−10
−5
0
5
10
$g_{(x)}$
0.5
0.0
−0.5
−1.0
−1.5
−2.0
x_1
x_2
10
8
6
4
2
0
0
2
4
6
8
10

(a)

(b)

(c)

(d)

(e)

(f)

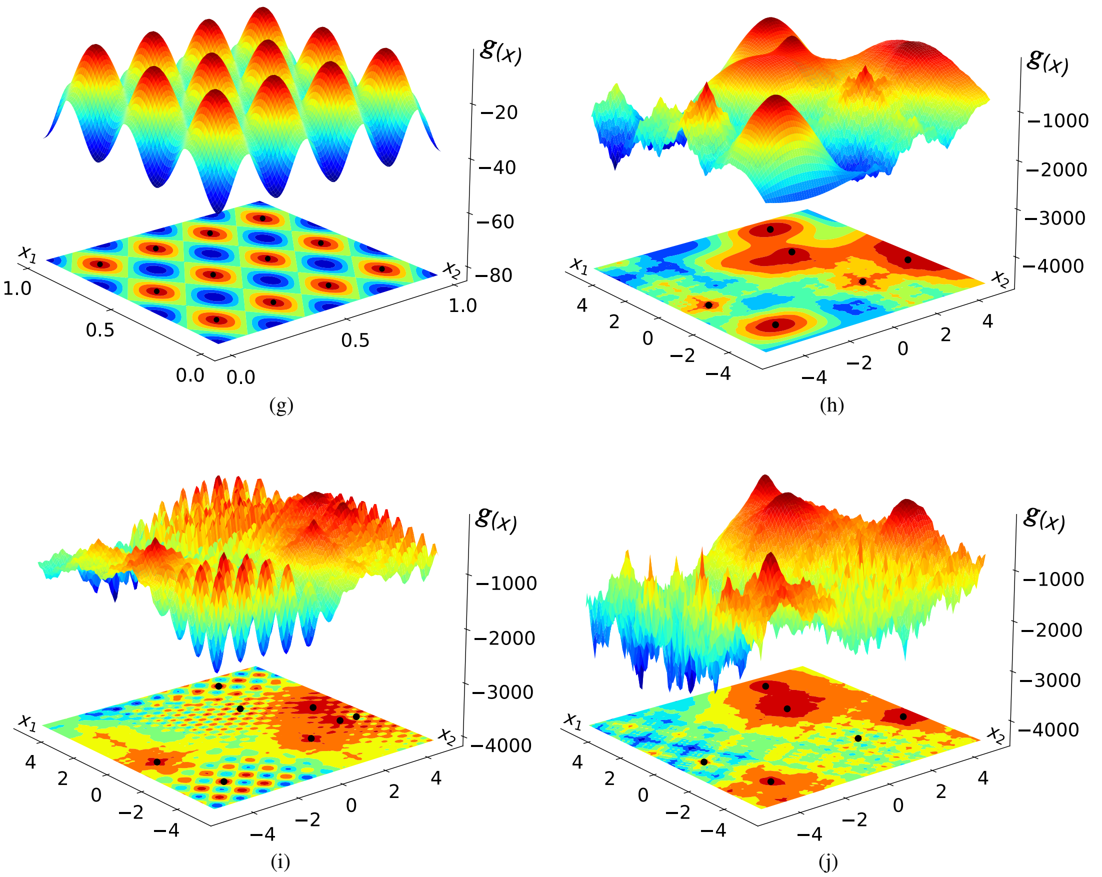


Fig. S.3. Fitness landscapes of ten functions. (a) F_1 (b) F_2 (c) F_3 (d) F_4 (e) F_6 (f) F_7 (g) F_{10} (h) F_{11} (i) F_{12} (j) F_{13}. The black points are optimal solutions.

TABLE S.II. Peak Ratio and Success Rate Obtained by TriDE at Five Accuracy Levels

Accuracy Level	1.0E-01		1.0E-02		1.0E-03		1.0E-04		1.0E-05	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
F_1	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000
F_2	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000
F_3	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000
F_4	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000
F_5	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000
F_6	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000
F_7	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000
F_8	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000
F_9	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000
F_{10}	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000
F_{11}	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000
F_{12}	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000
F_{13}	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000	**1.0000**	1.0000
F_{14}	**0.8600**	0.1600	**0.8600**	0.1600	**0.8600**	0.1600	0.8533	0.1200	0.8467	0.0800
F_{15}	**0.7500**	0.0000	**0.7500**	0.0000	**0.7500**	0.0000	**0.7500**	0.0000	**0.7500**	0.0000
F_{16}	**0.6667**	0.0000	**0.6667**	0.0000	**0.6667**	0.0000	**0.6667**	0.0000	**0.6667**	0.0000
F_{17}	**0.6800**	0.0000	**0.6800**	0.0000	**0.6800**	0.0000	**0.6800**	0.0000	**0.6800**	0.0000
F_{18}	**0.6667**	0.0000	**0.6667**	0.0000	**0.6667**	0.0000	**0.6667**	0.0000	**0.6667**	0.0000
F_{19}	**0.4850**	0.0000	**0.4850**	0.0000	**0.4850**	0.0000	**0.4850**	0.0000	**0.4850**	0.0000
F_{20}	**0.2400**	0.0000	**0.2400**	0.0000	**0.2400**	0.0000	0.2350	0.0000	0.2000	0.0000
Average	0.867	0.658	0.867	0.658	0.867	0.658	0.867	0.656	0.864	0.654

TABLE S.III. Peak Ratio and Success Rate Obtained by TriDE and the Other 14 Algorithms

ε=1.0E-04	FBK-DE		NBNC-PSO-ES		PMODE		ESPDE		NDC-DE	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
F_1	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000
F_2	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000
F_3	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000
F_4	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000
F_5	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000
F_6	0.990 +	0.820	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	0.999 +	0.980
F_7	0.813 +	0.000	0.967 +	0.140	0.672 +	0.000	0.963 +	0.360	0.881 +	0.000
F_8	0.824 +	0.000	0.808 +	0.000	0.616 +	0.000	0.880 +	0.000	0.941 +	0.000
F_9	0.425 +	0.000	0.540 +	0.000	0.324 +	0.000	0.729 +	0.000	0.458 +	0.000
F_{10}	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000
F_{11}	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000
F_{12}	0.935 +	0.480	**1.000** ≈	1.000	**1.000** ≈	1.000	0.930 +	0.560	0.941 +	0.520
F_{13}	**1.000** ≈	1.000	**1.000** ≈	1.000	0.953 +	0.720	0.793 +	0.080	**1.000** ≈	1.000
F_{14}	**0.907** –	0.460	0.847 +	0.100	0.800 +	0.000	0.727 +	0.000	0.833 +	0.160
F_{15}	0.730 +	0.000	0.738 +	0.000	0.750 ≈	0.000	0.730 +	0.000	**0.753** –	0.000
F_{16}	0.707 –	0.000	0.723 –	0.000	0.667 ≈	0.000	0.667 ≈	0.000	0.667 ≈	0.000
F_{17}	0.630 +	0.000	**0.718** –	0.000	0.405 +	0.000	0.685 –	0.000	0.633 +	0.000
F_{18}	**0.667** ≈	0.000	**0.667** ≈	0.000	0.500 +	0.000	0.660 +	0.000	**0.667** ≈	0.000
F_{19}	0.520 –	0.000	**0.538** –	0.000	0.245 +	0.000	0.455 +	0.000	0.412 +	0.000
F_{20}	0.450 –	0.000	0.483 –	0.000	0.250 –	0.000	0.265 –	0.000	0.355 –	0.000
+ (TriDE is better)	7		5		8		9		8	
≈ (TriDE is equal)	9		11		11		9		10	
– (TriDE is worse)	4		4		1		2		2	
ε=1.0E-04	DIDE		ANDE		NEA2		NMMSO		NetCDE$_{\text{MMOPs}}$	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
F_1	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000
F_2	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000
F_3	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000
F_4	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000
F_5	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000
F_6	**1.000** ≈	1.000	**1.000** ≈	1.000	0.950 +	0.380	0.992 +	0.880	**1.000** ≈	1.000
F_7	0.921 +	0.040	0.933 +	0.176	0.914 +	0.040	**1.000** ≈	1.000	0.947 +	0.255
F_8	0.692 +	0.000	0.944 +	0.078	0.240 +	0.000	0.899 +	0.020	0.999 +	0.902
F_9	0.571 +	0.000	0.512 +	0.000	0.581 +	0.000	0.978 +	0.120	0.511 +	0.000
F_{10}	**1.000** ≈	1.000	**1.000** ≈	1.000	0.988 +	0.860	**1.000** ≈	1.000	**1.000** ≈	1.000
F_{11}	**1.000** ≈	1.000	**1.000** ≈	1.000	0.960 +	0.760	0.990 +	0.940	0.984 +	0.902
F_{12}	**1.000** ≈	1.000	**1.000** ≈	1.000	0.840 +	0.160	0.993 +	0.940	0.904 +	0.471
F_{13}	0.987 +	0.920	0.686 +	0.000	0.957 +	0.740	0.983 +	0.900	0.667 +	0.000
F_{14}	0.773 +	0.020	0.667 +	0.000	0.807 +	0.060	0.720 +	0.000	0.667 +	0.000
F_{15}	0.748 ≈	0.000	0.632 +	0.000	0.718 +	0.000	0.632 +	0.000	0.630 +	0.000
F_{16}	0.667 ≈	0.000	0.667 ≈	0.000	0.673 –	0.000	0.660 +	0.000	0.667 ≈	0.000
F_{17}	0.593 +	0.000	0.397 +	0.000	0.695 –	0.000	0.468 +	0.000	0.480 +	0.000
F_{18}	**0.667** ≈	0.000	0.654 +	0.000	**0.667** ≈	0.000	0.650 +	0.000	**0.667** ≈	0.000
F_{19}	0.543 –	0.000	0.363 +	0.000	0.667 –	0.000	0.450 +	0.000	0.461 +	0.000
F_{20}	0.355 –	0.000	0.248 –	0.000	0.360 –	0.000	0.172 +	0.000	0.380 –	0.000
+ (TriDE is better)	6		9		10		13		10	
≈ (TriDE is equal)	12		10		6		7		9	
– (TriDE is worse)	2		1		4		0		1	
ε=1.0E-04	RS-CMSA-ES RS-CMSA-ES		RS-CMSA-ESII		AED-DDE		EMO-MMO		TriDE	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
F_1	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000**	1.000
F_2	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000**	1.000
F_3	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000**	1.000
F_4	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000**	1.000
F_5	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000**	1.000
F_6	0.999 +	0.980	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000**	1.000
F_7	0.998 +	0.920	**1.000** ≈	1.000	0.838 +	0.039	**1.000** ≈	1.000	**1.000**	1.000
F_8	0.875 +	0.000	0.997 +	0.760	0.747 +	0.000	**1.000** ≈	1.000	**1.000**	1.000
F_9	0.734 +	0.000	0.990 +	0.140	0.384 +	0.000	0.950 +	0.000	**1.000**	1.000
F_{10}	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000**	1.000
F_{11}	0.997 +	0.980	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000**	1.000
F_{12}	0.948 +	0.580	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000** ≈	1.000	**1.000**	1.000
F_{13}	0.997 +	0.980	0.993 +	0.960	0.686 +	0.000	0.997 +	0.980	**1.000**	1.000
F_{14}	0.803 +	0.100	0.850 +	0.100	0.667 +	0.000	0.733 +	0.060	0.853	0.120
F_{15}	0.745 +	0.000	0.750 ≈	0.000	0.637 +	0.000	0.595 +	0.000	0.750	0.000
F_{16}	0.667 ≈	0.000	**0.833** –	0.000	0.667 ≈	0.000	0.657 +	0.000	0.667	0.000
F_{17}	0.695 –	0.000	0.750 –	0.000	0.375 +	0.000	0.335 +	0.000	0.680	0.000
F_{18}	**0.667** ≈	0.000	**0.667** ≈	0.000	0.654 +	0.000	0.327 +	0.000	**0.667**	0.000
F_{19}	0.508 –	0.000	0.703 –	0.000	0.375 +	0.000	0.135 +	0.000	0.485	0.000
F_{20}	0.468 –	0.000	**0.618** –	0.000	0.250 –	0.000	0.080 +	0.000	0.235	0.000
+ (TriDE is better)	9		4		9		9		—	
≈ (TriDE is better)	8		12		10		11		—	
– (TriDE is better)	3		4		1		0		—	

TABLE S.IV. Effectiveness Investigation to Density Objective

ε=1.0E-04	TriDE		TriDE-1	
	PR	SR	PR	SR
F_1	**1.000**	1.000	**1.000**	1.000
F_2	**1.000**	1.000	**1.000**	1.000
F_3	**1.000**	1.000	**1.000**	1.000
F_4	**1.000**	1.000	**1.000**	1.000
F_5	**1.000**	1.000	**1.000**	1.000
F_6	**1.000**	1.000	**1.000**	1.000
F_7	**1.000**	1.000	**1.000**	1.000
F_8	**1.000**	1.000	**1.000**	1.000
F_9	**1.000**	1.000	**1.000**	1.000
F_{10}	**1.000**	1.000	**1.000**	1.000
F_{11}	**1.000**	1.000	0.960	0.760
F_{12}	**1.000**	1.000	0.990	0.920
F_{13}	**1.000**	1.000	0.973	0.840
F_{14}	**0.853**	0.120	0.807	0.060
F_{15}	**0.750**	0.000	0.748	0.000
F_{16}	**0.667**	0.000	**0.667**	0.000
F_{17}	**0.680**	0.000	0.635	0.000
F_{18}	**0.667**	0.000	**0.667**	0.000
F_{19}	**0.485**	0.000	0.425	0.000
F_{20}	**0.235**	0.000	0.180	0.000
Average	**0.867**	0.656	0.852	0.629

TABLE S.V. Effectiveness Investigation to RDM

ε=1.0E-04	TriDE		TriDE-2	
	PR	SR	PR	SR
F_1	**1.000**	1.000	**1.000**	1.000
F_2	**1.000**	1.000	**1.000**	1.000
F_3	**1.000**	1.000	**1.000**	1.000
F_4	**1.000**	1.000	0.740	0.320
F_5	**1.000**	1.000	**1.000**	1.000
F_6	**1.000**	1.000	**1.000**	1.000
F_7	**1.000**	1.000	**1.000**	1.000
F_8	**1.000**	1.000	**1.000**	1.000
F_9	**1.000**	1.000	0.990	0.160
F_{10}	**1.000**	1.000	**1.000**	1.000
F_{11}	**1.000**	1.000	0.667	0.000
F_{12}	**1.000**	1.000	0.750	0.000
F_{13}	**1.000**	1.000	0.667	0.000
F_{14}	**0.853**	0.120	0.667	0.000
F_{15}	**0.750**	0.000	0.480	0.000
F_{16}	**0.667**	0.000	0.653	0.000
F_{17}	**0.680**	0.000	0.220	0.000
F_{18}	**0.667**	0.000	0.167	0.000
F_{19}	**0.485**	0.000	0.125	0.000
F_{20}	**0.235**	0.000	0.125	0.000
Average	**0.867**	0.656	0.713	0.424

TABLE S.VI. Effectiveness Investigation on Different Merging Operations

ε=1.0E-04	TriDE		TriDE-KM		TriDE-NN	
	PR	SR	PR	SR	PR	SR
F_1	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_2	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_3	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_4	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_5	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_6	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_7	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_8	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_9	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_{10}	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_{11}	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_{12}	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_{13}	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_{14}	0.853	0.120	**0.867**	0.200	**0.867**	0.200
F_{15}	**0.750**	0.000	**0.750**	0.000	**0.750**	0.000
F_{16}	**0.667**	0.000	**0.667**	0.000	**0.667**	0.000
F_{17}	**0.680**	0.000	0.675	0.000	0.645	0.000
F_{18}	**0.667**	0.000	**0.667**	0.000	**0.667**	0.000
F_{19}	0.485	0.000	**0.490**	0.000	0.475	0.000
F_{20}	0.235	0.000	**0.250**	0.000	0.235	0.000
Average	0.867	0.656	**0.868**	0.660	0.865	0.660

TABLE S.VII. Effect Investigation to Different Population Size

ε=1.0E-04	TriDE		TriDE-3		TriDE-4		TriDE-5	
	PR	SR	PR	SR	PR	SR	PR	SR
F_1	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_2	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_3	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_4	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_5	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_6	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_7	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_8	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_9	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_{10}	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_{11}	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_{12}	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	0.995	0.960
F_{13}	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_{14}	0.853	0.120	0.867	0.200	**0.880**	0.320	0.840	0.040
F_{15}	**0.750**	0.000	0.725	0.000	**0.750**	0.000	**0.750**	0.000
F_{16}	**0.667**	0.000	**0.667**	0.000	**0.667**	0.000	**0.667**	0.000
F_{17}	**0.680**	0.000	0.630	0.000	0.645	0.000	0.670	0.000
F_{18}	**0.667**	0.000	0.653	0.000	**0.667**	0.000	**0.667**	0.000
F_{19}	**0.485**	0.000	0.340	0.000	0.405	0.000	0.460	0.000
F_{20}	**0.235**	0.000	0.110	0.000	0.170	0.000	0.185	0.000
Average	**0.867**	0.656	0.850	0.660	0.859	0.666	0.861	0.650

TABLE S.VIII. Effect Investigation on Different Number of Reference Points

ε=1.0E-04	TriDE		TriDE-6		TriDE-7		TriDE-8		TriDE-9	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
F_1	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_2	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_3	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_4	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_5	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_6	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_7	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_8	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_9	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_{10}	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_{11}	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_{12}	**1.000**	1.000	**1.000**	1.000	0.995	0.960	**1.000**	1.000	**1.000**	1.000
F_{13}	**1.000**	1.000	0.973	0.840	0.980	0.280	0.980	0.280	**1.000**	1.000
F_{14}	0.853	0.120	0.833	0.000	0.840	0.000	**0.860**	0.160	0.720	0.000
F_{15}	**0.750**	0.000	0.730	0.000	**0.750**	0.000	0.730	0.000	0.745	0.000
F_{16}	**0.667**	0.000	**0.667**	0.000	**0.667**	0.000	**0.667**	0.000	**0.667**	0.000
F_{17}	0.680	0.000	0.480	0.000	**0.690**	0.000	0.540	0.000	0.620	0.000
F_{18}	**0.667**	0.000	**0.667**	0.000	**0.667**	0.000	**0.667**	0.000	**0.667**	0.000
F_{19}	**0.485**	0.000	0.440	0.000	0.450	0.000	0.460	0.000	0.440	0.000
F_{20}	0.235	0.000	**0.300**	0.000	0.120	0.000	0.160	0.000	0.230	0.000
Average	**0.867**	0.656	0.855	0.642	0.857	0.612	0.853	0.622	0.854	0.650

Appendix A
Proof of Conflicts Between Objectives

With respect to the first two objectives as follows:

$$\begin{cases} f_1(\mathbf{x}) = \alpha(\mathbf{x}) + \beta(\mathbf{x}) = \frac{\sum_{j=1}^{RN} ||(\mathbf{x}-\mathbf{r}_j)|| \; - \min\limits_{t=1,\dots,PS} \sum_{j=1}^{RN} ||(\mathbf{x}_t-\mathbf{r}_j)||}{\max\limits_{t=1,\dots,PS} \sum_{j=1}^{RN} ||(\mathbf{x}_t-\mathbf{r}_j)|| \; - \min\limits_{t=1,\dots,PS} \sum_{j=1}^{RN} ||(\mathbf{x}_t-\mathbf{r}_j)||} + \frac{g(\mathbf{x}) \; - \min\limits_{t=1,\dots,PS} g(\mathbf{x})}{\max\limits_{t=1,\dots,PS} g(\mathbf{x}) \; - \min\limits_{t=1,\dots,PS} g(\mathbf{x})} \\ f_2(\mathbf{x}) = 1 - \alpha(\mathbf{x}) + \beta(\mathbf{x}) = 1 - \frac{\sum_{j=1}^{RN} ||(\mathbf{x}-\mathbf{r}_j)|| \; - \min\limits_{t=1,\dots,PS} \sum_{j=1}^{RN} ||(\mathbf{x}_t-\mathbf{r}_j)||}{\max\limits_{t=1,\dots,PS} \sum_{j=1}^{RN} ||(\mathbf{x}_t-\mathbf{r}_j)|| \; - \min\limits_{t=1,\dots,PS} \sum_{j=1}^{RN} ||(\mathbf{x}_t-\mathbf{r}_j)||} + \frac{g(\mathbf{x}) \; - \min\limits_{t=1,\dots,PS} g(\mathbf{x})}{\max\limits_{t=1,\dots,PS} g(\mathbf{x}) \; - \min\limits_{t=1,\dots,PS} g(\mathbf{x})} \end{cases} \tag{1}$$

we borrow the definition of Pareto-optimal solution to prove there exists conflicts between $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$.

Theorem: Supposing **u** is an optimal solution of an MMOP, after transformation of (1), **u must be a** Pareto-optimal solution.

Proof: If a decision vector **u** is an optimal solution of the given MMOP, but it is not the Pareto-optimal solution of (1), and then there should have a decision vector **v** that Pareto dominates **u**, which is shown as follows:

$$\begin{cases} f_1(\mathbf{v}) < f_1(\mathbf{u}) \\ f_2(\mathbf{v}) \leq f_2(\mathbf{u}) \end{cases} \quad \text{(2a)} \qquad \text{or} \qquad \begin{cases} f_1(\mathbf{v}) \leq f_1(\mathbf{u}) \\ f_2(\mathbf{v}) < f_2(\mathbf{u}) \end{cases} \quad \text{(2b)}$$

Regarding (2a), because the normalization method used in $\alpha(\mathbf{x})$ and $\beta(\mathbf{x})$ is the min-max scaling, the range of $\alpha(\mathbf{x})$ and $\beta(\mathbf{x})$ is $[0,1]$. As **u** is optimal solution, the value of $\beta(\mathbf{u})$ is zero. Thus, the expression can be simplified as follows:

$$\begin{cases} 0 \leq \beta(\mathbf{v}) < \alpha(\mathbf{u}) - \alpha(\mathbf{v}) \\ 0 \leq \beta(\mathbf{v}) \leq -[\alpha(\mathbf{u}) - \alpha(\mathbf{v})] \end{cases} \tag{3}$$

It is clear that the above two inequalities cannot be satisfied simultaneously, which implies that there does not exist **v** subject to **v** Pareto dominates **u**, and as a result, **u** is Pareto-optimal solution of (1). The situation in (2b) is the same.

Appendix B
Peak Ratio and Success Rate

The Peak Ratio (PR) and the success rate (SR) are calculated as follows:

$$\text{PR} = \frac{\sum_j^N N_{ago}^j}{NoG \cdot N} \qquad \text{SR} = \frac{N_{TS}}{N} \tag{4}$$

where N is the number of total trials and N_{ago}^j is the number of acceptable global optima within the corresponding accuracy level ε in the jth trial, NoG is the number of known global optima of the test function, N_{TS} is the number of total successful trials. If all known global optima are found in a single trial, such a trial is considered to be successful.

When determining whether the obtained solutions are acceptable or not, two kinds of the maximum permissible errors should be satisfied. In terms of objective function value, the user-defined accuracy level ε is used, and in terms of the decision space, the r presented in Table S.I for each test function is used. The source codes of evaluation criteria can be downloaded from: https://github.com/mikeagn/CEC2013

Appendix C
Niching Radius Calculation

Suppose that each individual is the center of a D-dimensional hypersphere. Therefore, the number of these hyperspheres is equal to the population size PS. To cover the entire decision space, each hypersphere must occupy $\frac{1}{PS}$ volume of the entire decision space. Let δ, V, and r denote the radius of a D-dimensional hypersphere occupied by an individual, the volume of the entire decision space, and the radius of a D-dimensional hypersphere of the entire decision space, respectively, and then δ can be calculated as follows:

$$V = cr^D \tag{5}$$

where c is a constant related to the D-dimensional hypersphere.

When the Euclidean distance is applied, r can be calculated by:

$$r = \frac{1}{2}\sqrt{\sum_{j=1}^{D}(x_j^u - x_j^l)^2} \tag{6}$$

where x_j^u and x_j^l are the upper and lower bound of x_j. As the variables are $0-1$ normalized before calculating the Euclidean distance, i.e., $x_j^u = 1$ and $x_j^l = 0$, r is given explicitly by:

$$r = \frac{\sqrt{D}}{2} \tag{7}$$

If the entire decision space is occupied by PS individuals, we may write

$$V = cr^D = PSc\delta^D \tag{8}$$

which yields

$$\delta = \frac{\sqrt{D}}{2\sqrt[D]{PS}} \tag{9}$$

If $\delta = \frac{\sqrt{D}}{2\sqrt[D]{PS}}$ is applied, this means that there is no overlap along the diagonal of two adjacent hyperspheres. According to the analyses and settings of the dynamic radius-niching technique [1], [2], to balance convergence and diversity, each niche should contain at least one of its nearest neighbors. Therefore, in this paper, δ_0 is 2δ, i.e., $\frac{\sqrt{D}}{\sqrt[D]{PS}}$.

References

[1] J. Gan and K. Warwick, "Dynamic niche clustering: a fuzzy variable radius niching technique for multimodal optimisation in GAs," in *Proc. IEEE Congr. Evol. Comput.*, vol. 1, 2001, pp. 215–222 vol. 1.

[2] W. Sheng, X. Wang, Z. Wang, Q. Li, and Y. Chen, "Adaptive memetic differential evolution with niching competition and supporting archive strategies for multimodal optimization," *Inform. Sci.*, 2021.