Parameter-Free Voronoi Neighborhood for Evolutionary Multimodal Optimization

Yu-Hui Zhang, Yue-Jiao Gong, *Member, IEEE*, Ying Gao, Hua Wang, and Jun Zhang, *Fellow, IEEE*

***Abstract*—Neighborhood information plays an important role in improving the performance of evolutionary computation in various optimization scenarios, particularly in the context of multimodal optimization. Several neighborhood concepts, i.e., index-based neighborhood, nearest neighborhood, and fuzzy neighborhood, have been studied and engaged in the design of niching methods. However, the use of these neighborhood concepts requires the specification of some problem-related parameters, which is difficult to determine without *a prior* knowledge. In this paper, we introduce a new neighborhood concept based on a geometrical construction called Voronoi diagram. The new concept offers two advantages at the expense of increasing the computational complexity to a higher level. It eliminates the need of additional parameters and it is more informative than the existing ones. The information provided by the Voronoi neighbors of an individual can be exploited to estimate the evolutionary state. Based on the information, we divide the population into three groups and assign each group a different reproduction strategy to support the exploration and exploitation of the search space. We show the use of the concept in the design of an effective evolutionary algorithm for multimodal optimization. The experiments have been conducted to investigate the performance of the algorithm. The results reveal that the proposed algorithm compare favorably with the state-of-the-art algorithms designed based on other types of neighborhood concepts.**

***Index Terms*—Evolutionary multimodal optimization, neighborhood information, niching technique, Voronoi diagram.**

Manuscript received June 28, 2018; revised November 28, 2018 and April 27, 2019; accepted June 2, 2019. Date of publication June 18, 2019; date of current version March 31, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61873095, Grant 61873097, and Grant U1701267, in part by the Science and Technology Planning Project of Guangzhou under Grant 201904010211, and in part by the Guangdong Natural Science Foundation Research Team under Grant 2018B030312003.

Y.-H. Zhang is with the School of Computer Science and Technology, Dongguan University of Technology, Dongguan 523808, China, and also with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China.

Y.-J. Gong and Y. Gao are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China, and also with the Guangdong Provincial Key Laboratory of Computational Intelligence and Cyberspace Information, South China University of Technology, Guangzhou 510006, China (e-mail: gongyuejiao@gmail.com).

H. Wang and J. Zhang are with the Institute for Sustainable Industries and Liveable Cities, University of Victoria, Melbourne, VIC 8001, Australia (e-mail: junzhang@ieee.org)

This paper has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org provided by the authors.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2019.2921830

I. Introduction

EVOLUTIONARY computation (EC) contains a family of computation paradigms inspired by the theory of evolution or the collective behavior of social animals. Different from traditional optimization methods, EC approaches maintain a population of individuals (candidate solutions) and iteratively refine them through genetic operators or social learning operators. The global search capability of EC approaches is mainly attributed to the interactions between individuals that share information about the fitness landscape. An individual can search the problem space more efficiently by cooperating with other individuals.

The scope of interactions between the individuals is determined by a communication graph. In the canonical EC approaches like genetic algorithm (GA) [1], differential evolution (DE) [2], evolutionary strategy (ES) [3], and particle swarm optimization (PSO) [4], a fully connected communication graph is commonly used, which suggests that all the individuals in the population have equal chances to exchange their genetic material. However, in nature, the interactions show a bias toward the pairs of individuals that are close to each other. Some groups of individuals might even be separated due to the geographic isolation. These observations give rise to the idea of localizing the evolution of individuals by imposing restrictions on their interactions. The concept of *neighborhood* comes naturally in this process. The neighbors of an individual are those connected directly with it in the communication graph. The local evolution of individuals can be realized by incorporating a neighborhood structure into the population.

In recent years, substantial research effort has been devoted to the enhancement of EC approaches by utilizing the neighborhood information and significant progress has been made in the research field. The most straightforward way to define neighborhoods for individuals is by constructing a communication topology. In PSO, the particles are organized according to a predefined topology and the neighborhood of a particle is established by its adjacent particles. Kennedy [5] has investigated a number of neighborhood topologies and has studied their effects on the performance of PSO. It is reported that large neighborhood topologies with dense interconnections exhibit fast convergence speed, but the swarm diversity may lose quickly. On the other hand, small neighborhood topologies with few interconnections show strong diversity maintenance capability. However, the convergence speed is slow down. Notice that a fixed communication topology is not appropriate for problems with different characteristics, some researchers made a step forward to study dynamic neighborhood structures [6]–[9].

The neighborhood topology has been generalized to the selection of parents in the mutation operators of DE. Das *et al.* [10] utilized the neighborhood concept to improve the DE/target-to-best/1/bin scheme and develop a DE variant called DE with global and local neighborhoods (DEGLs). They proposed two kinds of neighborhood models, a local model and a global model. A weight factor is introduced to achieve a tradeoff between the effects of the two models.

The neighbors of an individual can be defined by metric distance in the variable space. Epitropakis *et al.* [11] developed a proximity-based framework for DE that incorporates the information of neighboring solutions in the mutation operator to guide the evolution of the population. By favoring the parents in the vicinity of the mutated individual, the strategy promotes the efficiency of exploitation without substantially impairing the exploration capability.

It is also possible to define neighbors according to metric distance in the objective space [12]–[14]. Zhang and Li [12] proposed a multiobjective evolutionary algorithm (MOEA/D) that decomposes a multiobjective problem into a number of single objective subproblems using a set of weight vectors. The neighborhood relation is defined by the distances between the weight vectors and the neighborhood size is controlled by a parameter T. Each subproblem is optimized by utilizing information from its T neighboring solutions so as to reduce the computational complexity.

The above-mentioned developments of EC shed light on the importance of the neighborhood information in improving the search efficiency of the population-based algorithms. The appropriate use of the neighborhood information is essential for keeping a balance between exploitation and exploration. Generally speaking, the existing neighborhood concepts can be roughly divided into three categories, i.e., index-based neighborhood, nearest neighborhood, and fuzzy neighborhood. These concepts have been extensively used in the context of multimodal optimization, where an optimizer is required to find multiple optimal solutions for a given problem. When solving multimodal optimization problems, the neighborhood information plays an important role in inducing multiple convergence behaviors in EC. By restricting the interactions of individuals to a limited number of neighbors, the genetic drift phenomenon can be mitigated and the population diversity can be preserved. A number of methods known as *niching* have been developed and incorporated into EC approaches to facilitate the maintenance of population diversity and the formation of subpopulations. The niching methods motivate the individuals to evolve toward their nearby optimal solutions by modifying the reproduction and environmental selection operators using neighborhood information. In this way, multiple optimal solutions can be located by EC approaches within a single run. In the literature, all the three neighborhood concepts have been engaged in the development of niching methods [28]–[40].

Although the neighborhood structures are capable of enhancing the performance of EC in locating multiple optimal solutions, they are not easy to use due to the existence of problem-related parameters. The parameters are difficult to determine without knowing the specifics of the problem at hand. In this paper, instead of trying to devise an adaptive scheme for each of the concepts to dynamically tune the parameters, a parameter-free neighborhood concept is introduced. We then focus on the application of the new neighborhood concept in the design of an effective multimodal algorithm. The contribution of this paper lies in three aspects.

1) We systematically compare the existing neighborhood concepts and discuss their advantages and disadvantages. A taxonomy for evolutionary multimodal algorithms is presented with respect to the underlying neighborhood structures they are relying upon. Some recently proposed multimodal algorithms are reviewed and put into three different categories according to the taxonomy.
2) We introduce a new neighborhood concept called Voronoi neighborhood based on a space partitioning technique. The concept extends the current research by offering a new way to define neighbors for individuals. Compared with the existing concepts, the new concept is parameter-free and is more informative. Moreover, an approximation algorithm for finding Voronoi neighbors in high dimensional space is developed so that the concept can be put into practical use.
3) Motivated by the successful use of neighborhood information in the design of evolutionary multimodal algorithms, we propose a Voronoi neighborhood-based crowding DE (VNCDE) to solve multimodal optimization problems. An evolutionary state estimator (ESE) is designed and incorporated into VNCDE to extract useful information from the Voronoi neighborhood. The estimator divides the individuals to three different types according to their positions in the fitness landscape and VNCDE assigns each type of individuals a tailored search strategy to improve the search efficiency.

The experiments have been conducted on a set of benchmark problems to evaluate the performance of the proposed algorithm. The results suggest that the algorithm developed based on Voronoi neighborhood displays high performance. It is able to outperform state-of-the-art algorithms that build on the other neighborhood structures. The results also reveal the limitation of the Voronoi neighborhood. VNCDE suffers from high computational time overheads. This limitation needs to be taken into consideration before putting the proposed algorithm into practical use.

The remainder of this paper is organized as follows. Section II briefly reviews the evolutionary multimodal algorithms, the existing neighborhood concepts, and their applications in the algorithm design. The formal definition of Voronoi neighborhood is introduced in Section III. We highlight the distinct properties of the Voronoi neighborhood by comparing it with the existing ones. The means of finding Voronoi neighbors is also presented in this section. In Section IV, we demonstrate the use of the new concept in designing an effective multimodal algorithm. The experiments are conducted in Section V to evaluate the performance of the designed algorithm, with a detailed analysis of the numerical results. Finally, Section VI concludes this paper and provides some promising future research directions.

II. Background

In this section, the background of evolutionary multimodal optimization is first described. Then, we review the canonical DE and some classical algorithms to lay groundwork for the algorithm presented in Section IV. Subsequently, three neighborhood concepts, i.e., index-based neighborhood, nearest neighborhood, and fuzzy neighborhood, as well as their applications in the design of niching methods are discussed. The advantages and disadvantages of the neighborhood concepts are also included in the discussion.

A. Evolutionary Multimodal Optimization and Niching

Many problems encountered in scientific computing and engineering design are multimodal in nature, which means that there exist more than one setting of the decision variables that can achieve the optimal objective function value. When dealing with these problems, it is often desirable to find all the optimal solutions since they can provide additional benefits to decision makers. Some further insight into the problem can be obtained by examining the common structures of the solutions. Moreover, it is possible to combine the solutions to build a more robust system.

To handle the multimodal problems, traditional optimization methods have to run multiple times with different starting points. There is no guarantee that the methods converge to a different solution at each time. In comparison, EC approaches are well-suited for multimodal optimization owing to their population-based search mechanism. If the individuals in the population are committed to different sources of attraction, then multiple solutions can be located simultaneously. However, EC approaches are originally designed for global optimization. They generally converge to a single solution due to the genetic drift phenomena. To induce multiple convergence behavior, new methods known as *niching* have been developed. Niching methods modify the search behavior of individuals in order to support the formation of subpopulations. By incorporating niching methods, EC approaches have shown great promise in multiple solutions search. It is worth noting that nearly all the niching techniques encodes the neighborhood information in some form.

B. Differential Evolution

DE, proposed by Storn and Price [15], is one of the most popular EC paradigms. It is a promising candidate for multimodal optimization. In the optimization process, DE maintains a population of individuals. Let $P = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ denote the population. DE iteratively refines the individuals through mutation, crossover, and selection operators. Take $\boldsymbol{x}_i$ as an example, at each iteration, a mutant vector $\boldsymbol{v}_i$ is generated for $\boldsymbol{x}_i$ as follows:

$$\boldsymbol{v}_i = \boldsymbol{x}_{r1} + F \cdot (\boldsymbol{x}_{r2} - \boldsymbol{x}_{r3}) \tag{1}$$

where F is the scaling factor and $\boldsymbol{x}_{r1}$, $\boldsymbol{x}_{r2}$, and $\boldsymbol{x}_{r3}$ are different individuals randomly sampled from the population. After the mutation, a trial vector $\boldsymbol{u}_i$ is produced by performing crossover on the mutant vector $\boldsymbol{v}_i$ and the target vector $\boldsymbol{x}_i$

$$\boldsymbol{u}_{i,j} = \begin{cases} \boldsymbol{v}_{i,j}, & \text{if } \text{rand}_j < Cr \text{ or } j = j_{\text{rand}} \\ \boldsymbol{x}_{i,j}, & \text{otherwise} \end{cases} \tag{2}$$

Cr is the crossover probability and j_{rand} is a random number within the range $[1, D]$. Subsequently, competition takes place between $\boldsymbol{u}_i$ and $\boldsymbol{x}_i$. The one with better fitness will survive to the next iteration

$$\boldsymbol{x}_i = \begin{cases} \boldsymbol{u}_i, & \text{if } \boldsymbol{u}_i \text{ better than } \boldsymbol{x}_i \\ \boldsymbol{x}_i, & \text{otherwise.} \end{cases} \tag{3}$$

Das *et al.* [16] have provided an update survey of DE recently. Interested readers can refer to this paper for comprehensive information about the recent developments of DE.

C. Classical Niching Methods

Niching refers to the technique of finding and preserving multiple optimal solutions [17]–[25]. To avoid the individuals being confined to a single optimum, a substantial number of niching methods have been proposed in the literature. Most of the early niching methods focus on modifying the replacement mechanism of EC to offset the genetic drift caused by the greedy selection mechanism. Specifically, in the selection process, these niching methods not only take into account the fitness of the individuals but also their contribution to the population diversity. By doing so, individuals evolving toward different optimal solutions can both survive until the end of the search. Some classical niching methods that adhere to the principle include crowding [18] and speciation [20].

1) Crowding: Crowding [22] is a simple yet efficient niching method. It draws inspiration from the resource competition among animals living in the natural environment. The idea is that two close individuals must compete with each other for limited resources. For an offspring solution, a number of individuals are randomly sampled from the population. The sample size is controlled by a parameter called crowding factor (CF). The nearest neighbor to the offspring is extracted from the samples and is compared with the offspring. We replace the nearest neighbor with the offspring if the offspring has better fitness. A major problem with the crowding method is the replacement error. To address the problem, Mahfoud [23] proposed a deterministic variant that eliminates the sample size CF. An offspring is compared with its nearest neighbor in the entire population. In a subsequent study, Thomsen [18] integrated the crowding method with DE and developed a crowding DE (CDE).

2) Speciation: The speciation method [20] explicitly divides the population into multiple species according to the spatial correlation between individuals. Each species is formed around a dominant individual called species seed and is dedicated to the search of one optimum. Specifically, the species seed of a niche is first extracted from the population. Then, individuals whose distance to the species seed is less than a threshold radius are identified as members of the species. In later studies, Li embedded the speciation method into PSO and DE and proposed speciation PSO (SPSO) [24] and speciation DE (SDE) [25], respectively.

The above-mentioned niching methods are effective in preserving population diversity. However, they have difficulty in obtaining high accuracy solutions due to the random mating selection. When generating new solutions, the parents involved in reproduction may come from different subpopulations and may be far apart from each other. This is detrimental for the convergence of subpopulations, especially in the final stage of the search. Considering the balance between exploration and exploitation, Lynn and Suganthan [26] developed a heterogeneous comprehensive learning PSO (HCLPSO). HCLPSO divides the swarm population into two subpopulations. The two subpopulations are responsible for the exploration and exploitation tasks, respectively. In the optimization process, the comprehensive learning strategy is adopted to construct exemplars for the particles. Hui and Suganthan [27] proposed a novel ensemble and arithmetic recombination-based SDE (EARSDE) to solve multimodal optimization problems. The exploration capability is enhanced by a new speciation variant that incorporates the arithmetic recombination technique. Meanwhile, the exploitation of peaks is facilitated by neighborhood-based ensemble mutation strategies.

Another research avenue is to enhance the local exploitation ability through the use of neighborhood information. By integrating with neighborhood-based reproduction operators, the niching methods can be further enhanced. Many advanced multimodal algorithms, being characterized by different neighborhood structures, can be viewed as extensions of the early methods. In the following sections, we review some recently proposed algorithms according to the type of neighborhood concept they rely upon.

D. Niching Methods With Index-Based Neighborhood

The index-based neighborhood is the simplest neighborhood concept originally initiated with PSO. It is later extended to be used in DE, with the purpose of achieving a balance between exploration and exploitation for global optimization. The individuals in the population are arranged in a certain topology according to their indexes and the neighborhood relations are represented by the connections. Some recent studies have shown that the index-based neighborhood information is also useful in inducing niching behavior. To avoid the need of setting niching parameters, Li [28] proposed a ring topology PSO (rpso). Using the ring topology, the swarm is naturally divided into subswarms in the search process. Li [28] investigated four variants of rpso with overlapping and nonoverlapping neighborhoods and pointed out that *local memory* and *small communication topology* are two key factors for the success of PSO in multimodal optimization. Epitropakis *et al.* [29] incorporated the index-based neighborhood information into DE and put forward a family of new mutation strategies named DE/inrand. In the new strategies, the index-based neighbors of parents are involved in the generation of the offspring solutions.

The main advantages of the index-based neighborhood include its simplicity and computational efficiency. The neighbors of any given individual can be determined in a constant time. Consequently, incorporating this type of neighborhood information does not increase the time complexity of the algorithms. The drawback is that a suitable topology must be specified beforehand. Moreover, the index-based neighborhood ignores the spatial correlation between individuals. Two individuals adjacent in the topology are not intended to be close to each other in the search space. This might incur the problem of oscillation in the multimodal fitness landscape.

E. Niching Methods With K-Nearest Neighborhood

Another straightforward way to define neighbors is to consider the distances between individuals in the search space. More formally, the neighbors of an individual are defined as those whose distances to the individual are among the k smallest, where k is a parameter used to determine the neighborhood size. The nearest neighborhood is arguably the most widely adopted concept in multimodal optimization.

Li [30] developed a fitness Euclidean ratio-based PSO (FERPSO). Instead of heading toward the historical best position, each particle is attracted by its neighborhood point with the highest FER value. Qu *et al.* [31] proposed neighborhood mutation strategy to facilitate multiple convergences of DE. When generating a donor vector for individual $\boldsymbol{x}_i$, the parents are selected from the k closest individuals to $\boldsymbol{x}_i$. The mutation strategy is integrated with three DE variants, i.e., CDE, SDE, and ShDE. The resulting algorithms are termed NCDE, NSDE, and NShDE, respectively. To enhance the performance of niching PSO, Qu *et al.* [32] proposed a locally informed particle swarm optimizer (LIPS) that makes efficient use of the neighborhood information. In the velocity update process of a particle $\boldsymbol{x_i}$, several *local bests* in the vicinity of $\boldsymbol{x_i}$ are combined to guide its search. Gao *et al.* [33] put forward a clustering technique and a self-adaptive parameter control technique for multimodal optimization. The clustering technique divides the population into a number of subpopulations. In the clustering process, a reference point is randomly produced and the k nearest individuals to the reference point are combined to form a subpopulation. Gao *et al.* [33] integrated the two techniques with CDE and SDE and developed a self-adaptive cluster-based CDE (self-CCDE) and a self-adaptive cluster-based SDE (self-CSDE). Epitropakis *et al.* [34] devised two new mutation strategies (DE/nrand/1 and DE/nrand/2) to improve the niching ability of DE. The nearest neighbor of the parent is used as the basis in producing the offspring solution. In a later study, an enhanced algorithm (dADE/nrand/1) [35] was developed by incorporating a parameter adaptation scheme and a dynamic archive into DE/nrand/1. In addition to PSO and DE, the nearest neighborhood has also been employed in the design of multimodal estimation of distribution (MEDA) [36] and multimodal ant colony optimization (MACO) [37] algorithms.

The nearest neighborhood is generally more effective than the index-based neighborhood with respect to the ability of niche preservation. This is because it reduces the connections between individuals that are in different niches. The drawback of the concept is that additional computation cost is required. To find the nearest neighbors, we have to calculate the distance for each pair of individuals. This inevitably increases the time complexity. Moreover, a neighborhood size k needs to be specified. In multimodal fitness landscape, it might be the case that the basins of attraction are of different sizes and shapes and there does not exist a unified setting of k suitable for all the individuals in the population.

F. Niching Methods With Fuzzy Neighborhood

In the fuzzy neighborhood, given an individual $\boldsymbol{x}_i$, all other individuals in the population are potential neighbors of $\boldsymbol{x}_i$, but they vary in the membership degree. More specifically, each individual is assigned a selection probability according to its distance to $\boldsymbol{x}_i$. The smaller the distance, the higher the selection probability. The assignment strategy is generally formulated by a probabilistic model. An important point to recognize is that the nearest neighborhood can be viewed as a specialization of the fuzzy neighborhood, where the probability of selecting the k nearest neighbors is set to $1/k$ and the probability of selecting other individuals decays to zero.

Biswas *et al.* [38] proposed a parent-centric normalized neighborhood (PCNN) mutation operator. The operator adopts a probabilistic selection scheme to build a neighborhood for each individual. There is an inversely proportional relationship between the probability and the Euclidean distance. In the subsequent study, Biswas *et al.* [39] presented a local information sharing mechanism. Not only the neighborhood information but also the fitness of individuals is incorporated in the probabilistic model for parent selection. The underlying idea is to increase the probability of selecting elite individual in the vicinity of the target vector undergoing mutation. Zhang *et al.* [40] developed a fast niching technique based on the theory of locality sensitive hashing (LSH). Individuals in the population are mapped to a number of buckets using a family of hash functions. The information exchanges are restricted to individuals placed in the same bucket. By doing so, two close individual have a higher probability for recombination than those that are far apart.

The fuzzy neighborhood concept provides a fine-grained assessment of the relationships between individuals. It is more flexible than the nearest neighborhood and can be adapted to different multimodal fitness landscapes. Like the nearest neighborhood, we need to compute the distance matrix for the population. This imposes a burden on the time complexity of the algorithm. In addition, a probabilistic model has to be specified in order to assign membership degrees to the individuals. This requires more expertise and effort than specifying the neighborhood size k.

Besides the above-mentioned methods, there are niching methods that encode the neighborhood information indirectly. For example, some algorithms [41]–[43] transform multimodal problems into multiobjective problems and solve them using multiobjective optimization techniques. The neighborhood information is involved in the design of the second objective function to encourage the individuals to detect and exploit different optima in the search space. It is worth mentioning that the niching methods reviewed in this section are far from exhaustive. For more information about evolutionary multimodal optimization and niching methods, interested readers are advised to consult the survey papers [44] and [45].

III. Voronoi Neighborhood

In this section, we present the formal definition of Voronoi neighborhood. Then, an approximation algorithm for finding Voronoi neighbors in high dimensional space is developed. Finally, the characteristics of Voronoi neighborhood are highlighted through comparison with the existing neighborhood concepts.

A. Definition of Voronoi Neighborhood

The new neighborhood concept is defined based on the Voronoi diagram, which is one of the most fundamental structures in computational geometry. The Voronoi diagram (also known as Dirichlet tessellation) is proved to be very useful in a wide variety of fields, especially in computer science and engineering (e.g., cluster analysis, collision detection, and motion planning). The formal definition of Voronoi diagram is given as follows [46]. Let $P = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ denote a set of site points and $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ denote the Euclidean distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. The closure of a set A is denoted by $\bar{A}$ and the line segment from $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$ is denoted by $\overline{\boldsymbol{x}_i\boldsymbol{x}_j}$. $B(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the *bisector* of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ defined as

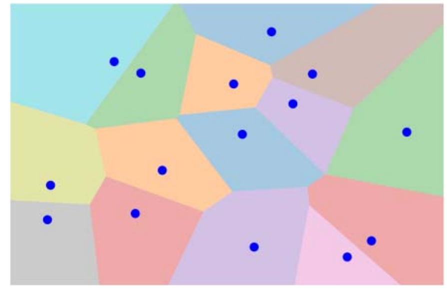


Fig. 1. Example of the Voronoi diagram for a set of randomly generated points.

$$B(\boldsymbol{x}_i, \boldsymbol{x}_j) = \{\boldsymbol{p} | d(\boldsymbol{p}, \boldsymbol{x}_i) = d(\boldsymbol{p}, \boldsymbol{x}_j)\}. \tag{4}$$

It is the perpendicular line through the center of the line segment $\overline{\boldsymbol{x}_i\boldsymbol{x}_j}$. It separates the halfplane $H(\boldsymbol{x}_i, \boldsymbol{x}_j)$ containing $\boldsymbol{x}_i$ and the halfplane $H(\boldsymbol{x}_j, \boldsymbol{x}_i)$ containing $\boldsymbol{x}_j$. The halfplane $H(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is defined according to the following formula:

$$H(\boldsymbol{x}_i, \boldsymbol{x}_j) = \{\boldsymbol{p} | d(\boldsymbol{p}, \boldsymbol{x}_i) < d(\boldsymbol{p}, \boldsymbol{x}_j)\}. \tag{5}$$

The *Voronoi region* (or *Voronoi cells*) of $\boldsymbol{x}_i$ with respect to P is defined as

$$VR(\boldsymbol{x}_i, P) = \bigcap_{\boldsymbol{x}_j \in P, i \neq j} H(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{6}$$

Finally, the *Voronoi diagram* of P is defined as

$$V(P) = \bigcup_{\boldsymbol{x}_i, \boldsymbol{x}_j \in P, i \neq j} \overline{VR(\boldsymbol{x}_i, P)} \cap \overline{VR(\boldsymbol{x}_j, P)}. \tag{7}$$

For brevity, we may write R_i for $VR(\boldsymbol{x}_i, P)$. According to the definition, each Voronoi region R_i is the intersection of $N - 1$ halfplanes containing $\boldsymbol{x}_i$. Hence, the Voronoi regions are disjoint convex polygons. The common boundary of two Voronoi regions is called a *Voronoi edge*. If a Voronoi edge e borders the regions R_i and R_j, then $e \subseteq B(\boldsymbol{x}_i, \boldsymbol{x}_j)$ holds. The endpoints of Voronoi edges are called *Voronoi vertices*, they belong to the common boundary of three or more Voronoi regions. The numbers of vertices and edges of the polygon are determined by its surrounding site points. Provided N site points, there are at most $2N - 5$ Voronoi vertices and $3N - 6$ Voronoi edges. An example of the Voronoi diagram is given in Fig. 1. From the figure, it can be observed that the entire proximity information about the site points has been included in an explicit manner. Another thing worth noting is that the Voronoi diagram of a set of points is dual to the Delaunay triangulation for the same set of points. Given a population of individuals, the Voronoi neighborhood of an individual $\boldsymbol{x}_i$ is defined as follows.

Definition: An individual $\boldsymbol{x}_j$ is called a Voronoi neighbor of $\boldsymbol{x}_i$ if their associated Voronoi cells are adjacent (have a common edge). The Voronoi neighborhood of $\boldsymbol{x}_i$ is defined as the set containing all the Voronoi neighbors of $\boldsymbol{x}_i$.

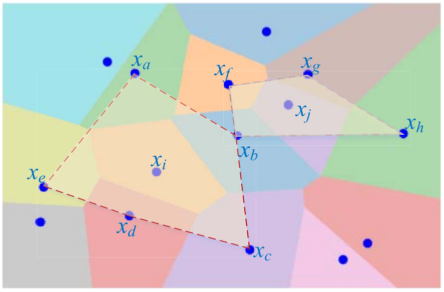


Fig. 2. Illustration of the definition of Voronoi neighborhood.

Algorithm 1 Voronoi Neighborhood Construction for $\boldsymbol{x}_i$

Initialize the neighbor list L_i to be empty;
for $\boldsymbol{x}_j \in P \backslash \{\boldsymbol{x}_i\}$:
  $\boldsymbol{m}_{ij} = (\boldsymbol{x}_i + \boldsymbol{x}_j)/2$;
  $mindis \leftarrow \infty$
  for $\boldsymbol{x}_k \in P \backslash \{\boldsymbol{x}_i, \boldsymbol{x}_j\}$:
    if $mindis > dis(\boldsymbol{x}_k, \boldsymbol{m}_{ij})$ **then:**
      $mindis = dis(\boldsymbol{x}_k, \boldsymbol{m}_{ij})$;
    end if
  end for
  if $mindis \geq dis(\boldsymbol{x}_i, \boldsymbol{m}_{ij})$ **then:**
    $L_i = L_i \cup \{\boldsymbol{x}_j\}$;
  end if
end for

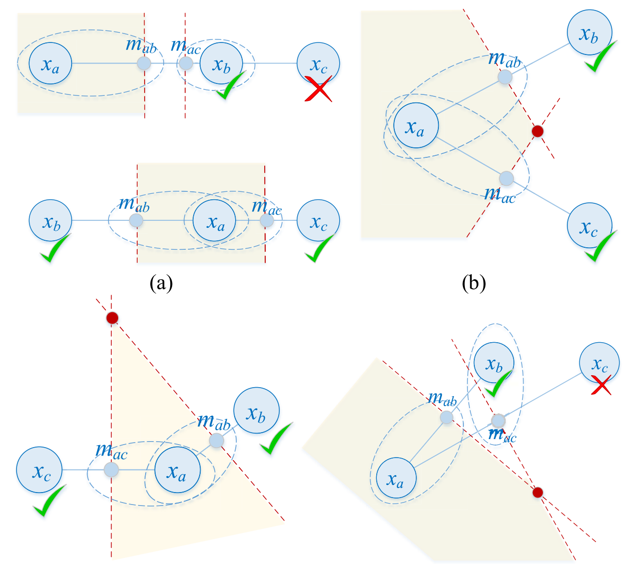


Fig. 3. Illustration of the approximation algorithm for finding Voronoi neighbors. (a) Correctness in 1-D space. (b) and (c) Two typical cases in 2-D space. (d) Type II error.

To illustrate the definition, Fig. 2 plots the Voronoi neighborhoods of two individuals located in the middle of the diagram ($\boldsymbol{x}_i$ and $\boldsymbol{x}_j$). It is easy to see that the Voronoi neighborhoods of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are given by $L_i = \{\boldsymbol{x}_a, \boldsymbol{x}_b, \boldsymbol{x}_c, \boldsymbol{x}_d, \boldsymbol{x}_e\}$ and $L_j = \{\boldsymbol{x}_b, \boldsymbol{x}_f, \boldsymbol{x}_g, \boldsymbol{x}_h\}$, respectively.

B. Approximation Algorithm for Finding Voronoi Neighbors

There exist several polynomial time algorithms for computing the Voronoi diagram in 2-D space. For an arbitrary set of site points, Shamos [47] showed that the optimal time bound is $O(N\log N)$. The divide and conquer algorithm [48] and the plane-sweep algorithm [49] are examples of the asymptotic optimal algorithms that match the time bound. However, these algorithms cannot be generalized to higher dimensional space. The Bowyer–Watson algorithm [50], [51], which is developed by Bowyer and Watson independently, is able to generate the Delaunay triangulation in any number of dimensions. The Voronoi diagram can be obtained from the Delaunay triangulation according to the duality relation between them. The time complexity of the algorithm is $O(a_D N^{(1+1/D)} + b_D N)$, where a_D and b_D are two constants depend on the dimension D. Simulations conducted in [50] showed that the two constants increase rapidly with the increase of D. Brown [52] perceived the possibility of transforming the problem into the convex finding problem. The method is characterized by its good generalization ability. However, determining high dimensional Voronoi diagram is computationally very expensive by means of the transformation. In practice, the problem dimensionality is generally larger than two. Hence, the existing methods cannot be applied since their execution time grows exponentially with the dimensionality. To address the problem, we devise an efficient and practical approximation algorithm capable of finding the Voronoi neighbors in an acceptable time.

The pseudocode of the approximation algorithm is given in Algorithm 1. The main question is how to determine whether two individuals are adjacent with respect to the definition of Voronoi neighborhood. Suppose that $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are two individuals being considered. We first calculate the midpoint ($\boldsymbol{m}_{ij}$) of the segment connecting $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Then, the distances from the midpoint to all other individuals are computed and the minimum distance (*mindis*) is recorded. Subsequently, we test whether the distance between $\boldsymbol{m}_{ij}$ and $\boldsymbol{x}_i$ is smaller than *mindis*. If the answer is positive, the two individuals are recognized to be Voronoi neighbors. If the answer is negative, $\boldsymbol{x}_j$ is excluded from the neighbor list of $\boldsymbol{x}_i$. In the second case, since there exists an individual closer to the midpoint, it is very likely that the Voronoi cells of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are separated by another cell that lies between them.

Fig. 3 illustrates how the approximation algorithm works. In the figure, the blue circles are used to represent individuals (site points). The notation $\boldsymbol{m}_{ab}$ indicates the midpoint of individual $\boldsymbol{x}_a$ and individual $\boldsymbol{x}_b$. Similarly, $\boldsymbol{m}_{ac}$ is the midpoint of $\boldsymbol{x}_a$ and $\boldsymbol{x}_c$. Bisectors of the individuals are shown in red dashed lines. The ellipse with dashed borders points out the nearest individual to the midpoint. The notation "$\surd$" indicates that the corresponding individual is recognized as a Voronoi neighbor of $\boldsymbol{x}_a$ and the notation "$\times$" indicates the opposite.

There are several situations indistinguishable to the approximation algorithm and this leads to approximation errors. Generally speaking, the approximation errors can be categorized into two groups, i.e., false positive error (Type I error) and false negative error (Type II error). A Type I error means that two separated individuals are mistakenly identified as neighbors, while a Type II error refers to the exclusion of an adjacent individual from the neighbor list. It is easy to infer that the approximation algorithm is free from the Type I errors. In 1-D space, the algorithm can always return exact Voronoi neighbors, as illustrated in Fig. 3(a). In higher

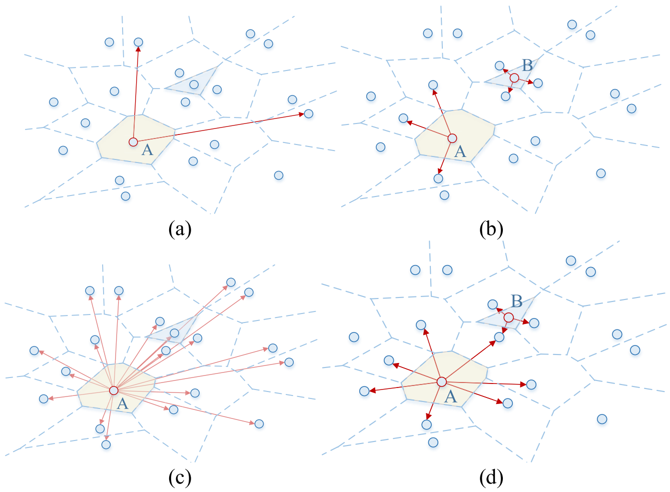


Fig. 4. Illustration of the four neighborhood concepts. (a) Index-based neighborhood. (b) Nearest neighborhood. (c) Fuzzy neighborhood. (d) Voronoi neighborhood.

TABLE I
CHARACTERISTICS OF THE FOUR NEIGHBORHOOD CONCEPTS

Type	PF	DIE	SIE	Symmetric	Complexity
Index-based neighborhood	No①	No	No	Yes	$O(1)$
Nearest neighborhood	No②	Yes	No	No	$O(D{\cdot}N)$
Fuzzy neighborhood	No③	Yes	No	No	$O(D{\cdot}N)$
Voronoi neighborhood	Yes	Yes	Yes	Yes	$O(D{\cdot}N^2)$

PF: Parameter-free.
DIE: Distance information embedded.
SIE: Structure information embedded.
① Communication topology. ② Neighborhood size. ③ Probabilistic model.

dimensional space, some Type II errors may occur. Fig. 3(b) and (c) shows two examples where the Voronoi neighbors are correctly identified (both $\boldsymbol{x}_b$ and $\boldsymbol{x}_c$ are Voronoi neighbors of $\boldsymbol{x}_a$). An example of the Type II error is demonstrated in Fig. 3(d) ($\boldsymbol{x}_c$ is a Voronoi neighbor of $\boldsymbol{x}_a$ but it is falsely rejected due to the existence of $\boldsymbol{x}_b$). It can be observed that the line segments $\boldsymbol{x}_a\boldsymbol{x}_b$ and $\boldsymbol{x}_a\boldsymbol{x}_c$ are oriented in roughly the same direction and $\boldsymbol{x}_b$ plays a similar role as $\boldsymbol{x}_c$. Therefore, the information loss caused by the exclusion of $\boldsymbol{x}_c$ is not very significant.

The running time of the approximation algorithm can be estimated directly from the pseudocode. There are totally $N-1$ midpoints given N individuals. For each midpoint, $N-1$ distance calculations are required. Note that a single distance calculation takes $O(D)$ time, the overall time complexity of the algorithm is $O(D \cdot N^2)$.

C. Comparison With Existing Neighborhood Concepts

Fig. 4 illustrates the four neighborhood concepts and Table I summarizes their features. In the figure, the red arrow lines are used to indicate the neighborhood relation. It means that the end point is one of the neighbors of the starting point for a specific neighborhood definition. The index-based neighborhood is efficient in terms of processing complexity, but it ignores the spatial correlation between individuals. In comparison, the nearest neighborhood is more straightforward and intuitive, and the fuzzy neighborhood is more generalized and flexible. They all require the users to specify some parameters, which may be difficult to decide (i.e., the communication topology in index-based neighborhood, the size parameter in nearest neighborhood, or the probabilistic model in the fuzzy neighborhood). Unlike these concepts, the Voronoi neighborhood is parameter-free. The neighbors of individuals are uniquely determined by the population distribution. Moreover, the Voronoi neighborhood is more informative than the existing ones. Not only the distance information but also the structural information is embedded in the neighborhood construction. Here, the structural information refers to the information about how the individuals are arranged in the search space. The Voronoi neighborhood provides a sort of structural information that tells us the surrounding neighbors around each individual.

In some special cases, the Voronoi neighborhood may be the same as the nearest neighborhood [e.g., the individual in the Voronoi cell B shown in Fig. 4(b) and (d)], but at other times they differ considerably [e.g., the individual in Voronoi cell A shown in Fig. 4(b) and (d)]. Another difference worth noting is that the Voronoi neighborhood relation is symmetric while the nearest neighborhood relation is nonsymmetric. In the definition of nearest neighborhood, it may happen that $\boldsymbol{x}_i$ is one of the k-nearest neighbors of $\boldsymbol{x}_j$ but $\boldsymbol{x}_i$ is not in the nearest neighbor list of $\boldsymbol{x}_i$.

Although the Voronoi neighborhood offers some advantages over the existing concepts, it is not without defects. The time complexity of finding Voronoi neighbors is higher than those of the other three types of neighbors. We need to calculate the distance for $O(N^2)$ pairs of points. Generally, the distances are computed in a sequential manner. Since the distance calculation for each pair of points is an independent task, it is possible to reduce the running time by distributing all the calculation tasks over multiple CPU cores. Nevertheless, the best way to address the complexity issue is to develop more advanced approximation algorithms capable of finding Voronoi neighbors in $O(N\lg N)$ time or in linear time.

IV. VORONOI NEIGHBORHOOD BASED CROWDING DE FOR MULTIMODAL OPTIMIZATION

In this section, we demonstrate the use of the Voronoi neighborhood in the design of an effective multimodal optimization algorithm. During the search process, the Voronoi neighbors of the individuals are identified using the approximation algorithm. Then, the evolutionary states of the individuals are estimated by examining their correlations with the Voronoi neighbors and the population is divided into three groups accordingly. Each group is assigned a tailored search strategy. Lastly, a VNCDE is developed by integrating the multiple-strategy search mechanism with the crowding selection technique.

A. Evolutionary State Estimator

To extract useful information from the Voronoi neighborhood, we propose an ESE. The pseudocode of ESE is presented in Algorithm 2. From Fig. 2, it can be observed that the individuals are generally surrounded by their Voronoi

Algorithm 2 Evolutionary State Estimation for $\boldsymbol{x}_i$

Initialize the set of learning exemplars V to be empty;
for $\boldsymbol{x}_k \in L_i$:
    if $\boldsymbol{x}_k$ better than $\boldsymbol{x}_i$ **then:**
        $V = V \cup \{\boldsymbol{x}_k\}$;
    end if
end for
if V is empty **then:**
    Attach the "*dominator*" label to $\boldsymbol{x}_i$;
else:
    compute θ according to (8);
    if $\theta < \pi/2$ **then:**
        Attach the "*challenger*" label to $\boldsymbol{x}_i$;
    else:
        Attach the "*explorer*" label to $\boldsymbol{x}_i$;
    end if
end if

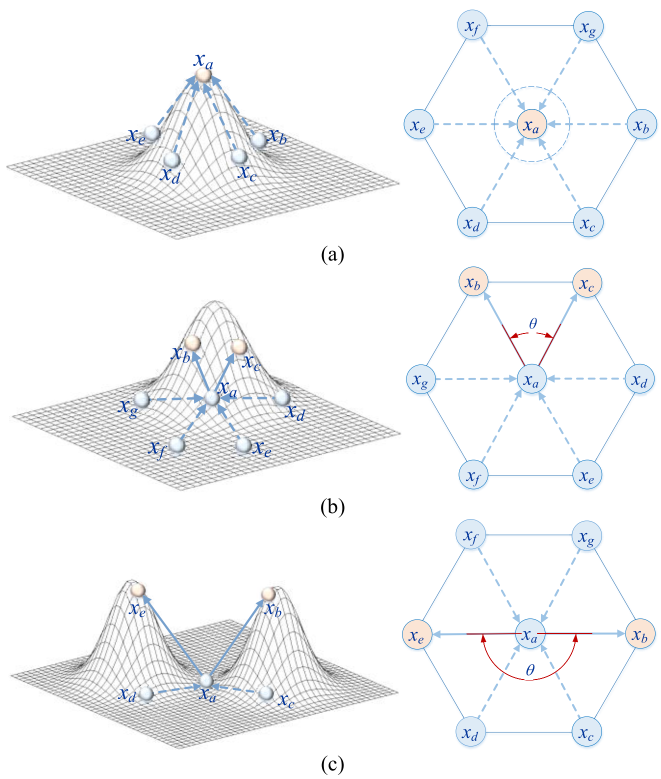


Fig. 5. Illustration of the three types of individuals. (a) *Dominator*. (b) *Challenger*. (c) *Explorer*.

neighbors (except for some individuals located in the corner of the search space). The positions of the individuals in the multimodal landscape can be estimated by exploiting the observation. Fig. 5 illustrates the basic principle. Suppose that we are handling maximization problems. For an individual $\boldsymbol{x}_a$, a comparison is made between its fitness and those of its Voronoi neighbors. The neighbors with better fitness are collected in a set V. Then, the average of the angles formed by the rays $\{\boldsymbol{x}_a\boldsymbol{x}_k | \boldsymbol{x}_k \in V\}$ is computed. Specifically, the average angle θ is determined by the following formula:

$$\theta = \frac{\sum_{\boldsymbol{x}_i,\boldsymbol{x}_j \in V} \angle \boldsymbol{x}_i\boldsymbol{x}_a\boldsymbol{x}_j}{(m-1)m/2} \tag{8}$$

where

$$\angle \boldsymbol{x}_i\boldsymbol{x}_a\boldsymbol{x}_j = \arccos\left(\frac{(\boldsymbol{x}_i - \boldsymbol{x}_a)(\boldsymbol{x}_j - \boldsymbol{x}_a)^T}{\|\boldsymbol{x}_i - \boldsymbol{x}_a\| \|\boldsymbol{x}_j - \boldsymbol{x}_a\|}\right) \tag{9}$$

and m represents the number of individuals in V. There are three possible scenarios for the comparison.

In the first scenario, the set V is empty and no surrounding individuals have better fitness than the individual $\boldsymbol{x}_a$, as demonstrated in Fig. 5(a). This suggests that the individual is close to a peak in the search space. The individual is therefore labeled as *dominator*.

In the second scenario, the set V contains several superior individuals and the average angle θ is smaller than $\pi/2$, as shown in Fig. 5(b). Because the learning directions are inclined at a small acute angle, it is likely that $\boldsymbol{x}_a$ is climbing a hill. The individual is in a proper position to challenge the dominator of its niche and can potentially be improved by moving toward the directions indicated by its Voronoi neighbors. Hence, a *challenger* label is attached to the individual.

In the third scenario, there are multiple Voronoi neighbors that have better fitness than $\boldsymbol{x}_a$ and the average angle is equal to or larger than $\pi/2$, as depicted in Fig. 5(c). This indicates that the individual locates in a valley and there exist multiple moving directions that can improve the fitness of $\boldsymbol{x}_a$. Since $\boldsymbol{x}_a$ has not been restricted to any optima in the search space, a more exploratory search can be conducted by the individual without sacrificing the convergence speed. Therefore, the individual is assigned a label of *explorer*.

By performing systematic comparisons, the ESE divides the individuals into three different groups, namely, *dominators*, *challengers*, and *explorers*, according to their role in the population.

B. Voronoi Neighborhood-Based Crowding DE

A VNCDE is designed to tackle multimodal problems by making use of the information extracted by the ESE. The detailed procedures of VNCDE are provided in Algorithm 3. At the beginning of the algorithm, a population of individuals are randomly sampled from the search space. Then, the individuals are iteratively updated via genetic operators. The detailed procedures are described as follows. For each individual in the population, the Voronoi neighbors are first identified using the approximation algorithm shown in Algorithm 1. Next, we estimate the evolutionary state of the individual using the ESE algorithm shown in Algorithm 2. According to the estimated evolutionary state, a tailored search strategy is employed to generate an offspring solution. After the offspring solution has been produced and evaluated, VNCDE compares the offspring with the nearest individual in the population. The nearest individual will be replaced if the offspring has better fitness. The above procedures are repeated until the termination criterion is satisfied.

The main idea of the algorithm is to assign the most suitable search strategy to each type of individuals so that the peaks can be located with improved efficiency and the multiple convergence ability of CDE can be enhanced. Specifically, the offspring solutions are generated with three different strategies.

For individuals classified as *dominators*, no individual in their Voronoi neighborhoods has higher fitness values. It is

Algorithm 3 VNCDE

Generate an initial population of individuals $P = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ by uniformly and randomly sample N individuals in the search space;
Evaluate the fitness of the individuals in the initial population;
$FEs = N$;
while $FEs < MaxFEs$ **do:**
  for $i = 1$ to N:
    Find the Voronoi neighbors of $\boldsymbol{x}_i$ using **Algorithm** 1;
    Use ESE to estimate the evolutionary state of $\boldsymbol{x}_i$;
    if $\boldsymbol{x}_i$ is classified as *dominator* **then:**
      Generate an offspring solution $\boldsymbol{u}_i$ with the local search strategy (10);
    end if
    if $\boldsymbol{x}_i$ is classified as *challenger* **then:**
      Generate an offspring solution $\boldsymbol{u}_i$ with the directional search strategy (11);
    end if
    if $\boldsymbol{x}_i$ is classified as *explorer* **then:**
      Generate an offspring solution $\boldsymbol{u}_i$ with the exploratory search strategy (12);
    end if
    Evaluate the fitness of $\boldsymbol{u}_i$;
    $FEs = FEs + 1$;
    Find the individual $\boldsymbol{x}_j$ in the population that has the smallest distance to $\boldsymbol{u}_i$;
    if the fitness of $\boldsymbol{u}_i$ is better than $\boldsymbol{x}_j$ **then:**
      Replace $\boldsymbol{x}_j$ with $\boldsymbol{u}_i$;
    end if
  end for
end while

probably that each dominator is close to a peak in the search space. Therefore, a Gaussian-based local search strategy that makes small modifications to the individuals are employed, with the intention that the dominators can move to their nearby peaks. The offspring solution $\boldsymbol{u}_i$ is produced by adding small perturbations to the parent $\boldsymbol{x}_i$, as formulated in

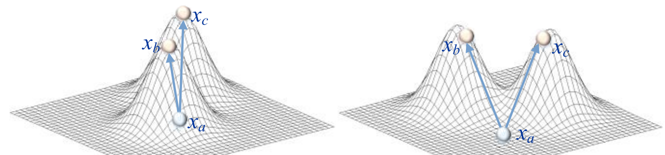
$$\boldsymbol{u}_{i,j} = \boldsymbol{x}_{i,j} + G(0, \sigma), \quad \text{for } \boldsymbol{j} = 1, 2, \ldots, D. \tag{10}$$

$G(0, \sigma)$ denotes a random number drawn from the Gaussian distribution with mean zero and standard deviation σ. The standard deviation σ is set to 1E-r to locate peaks with high-accuracy requirement and to increase the algorithm's ability for fine tuning (r is from a discrete uniform distribution with mean five).

For individuals classified as *challengers*, there are several individuals with higher fitness in their Voronoi neighborhoods. The difference vectors between the challengers and the better individuals point to roughly the same direction. It is probably that the challengers are in positions halfway up a hill. Therefore, a directional search strategy "DE/current-to-*nbest*" is employed so that the challengers can learn from the elite individuals in their corresponding region of attractions. In this way, the convergence rate of subpopulations can be improved. The learning exemplar $\boldsymbol{x}_{nbest}$ is selected from the set of Voronoi neighbors superior to $\boldsymbol{x}_i$ (roulette wheel selection based on normalized fitness). The offspring solution $\boldsymbol{u}_i$ is generated in the following manner:

$$\boldsymbol{u}_{i,j} = \boldsymbol{x}_{i,j} + F \cdot \left(\boldsymbol{x}_{nbest,j} - \boldsymbol{x}_{i,j}\right) + F \cdot \left(\boldsymbol{x}_{r1,j} - \boldsymbol{x}_{r2,j}\right) \tag{11}$$

where $\boldsymbol{x}_{r1}$ and $\boldsymbol{x}_{r2}$ are the two neighbors distinct from $\boldsymbol{x}_{nbest}$.



(a) (b)

Fig. 6. Two possible scenarios where the angle is smaller than $\pi/2$ but the learning exemplars located in different regions of attraction. (a) One peak behind another. (b) Two close peaks.

The exemplar $\boldsymbol{x}_{nbest}$ has a high probability to be in the same peak region as $\boldsymbol{x}_i$. However, it is worth noting that there are two scenarios not in line with the case shown in Fig. 5(b). The two scenarios are illustrated in Fig. 6. The first scenario is that one peak is behind another peak in the moving direction of individuals. The second scenario is that two peaks are close to each other in the search space. As shown in Fig. 6, the angles are less than $\pi/2$ but the learning exemplars lie in different peaks. In these scenarios, the DE/current-to-*nbest* strategy will be employed by VNCDE as well. Whichever the neighbor chosen, the difference vector between $\boldsymbol{x}_a$ and *nbest* will point to approximately the same direction after adding another difference term $\boldsymbol{x}_{r1} - \boldsymbol{x}_{r2}$. It is expected that the generated mutant vector brings $\boldsymbol{x}_a$ closer to one of the peaks. Hence, DE/current-to-*nbest* is still a suitable strategy to use.

For individuals classified as *explorers*, they are inferior to some of the individuals in their Voronoi neighborhoods. The difference vectors between the explorers and the superior neighbors point to opposite directions. It is probably that the explorer are located in valleys or in outer ranges of a peak region. Therefore, the global search strategy "DE/rand/1" is utilized to motivate them to explore other search regions. By doing so, the exploration capability of the algorithm can be enhanced without impairing the exploitation capability. The jth dimension of the offspring solution is computed according to the following formula:

$$\boldsymbol{u}_{i,j} = \begin{cases} \boldsymbol{x}_{r1,j} + F \cdot \left(\boldsymbol{x}_{r2,j} - \boldsymbol{x}_{r3,j}\right), & \text{if rand}(0, 1) \\ & \leq Cr \text{ or } j = j_{\text{rand}} \\ \boldsymbol{x}_{i,j}, & \text{otherwise} \end{cases} \tag{12}$$

where $\boldsymbol{x}_{r1}$, $\boldsymbol{x}_{r2}$, and $\boldsymbol{x}_{r3}$ are the individuals randomly sampled from the Voronoi neighborhood of $\boldsymbol{x}_i$.

C. Complexity Analysis

The proposed algorithm contains five major steps, namely, the initialization, the identification of Voronoi neighbors, the ESE procedure, the reproduction procedure, and the environmental selection procedure. The initialization runs in $O(D \cdot N)$ time. It is only executed once at the beginning of the algorithm. The other four steps are in the main loop of VNCDE and are executed repeatedly. Both the ESE procedure and the reproduction procedure require $O(D \cdot N)$ time. The time spent on the environmental selection is $O(D \cdot N^2)$. The procedure used to find Voronoi neighbors is the most time-consuming step that dominates other terms. According to the analysis conducted in the previous section, the running time of the procedure is $O(D \cdot N^3)$. Therefore, the overall time complexity of VNCDE is $O(D \cdot N^3)$ per iteration.

TABLE II
ALGORITHMS IN COMPARISON

No.	Category	Algorithm	Description
1	Index-based neighborhood	DE/inrand/1 [29]	DE/nrand/1 with index-based neighborhoods
2		r2pso [28]	A local best PSO with the ring topology, each particle interacts with its immediate neighbor on its right
3		r3pso [28]	A local best PSO with the ring topology, each particle interacts with its immediate neighbor on its left and right
4		r2pso-lhc [28]	r2pso without overlapping neighborhoods
5		r3pso-lhc [28]	r3pso without overlapping neighborhoods
6	Nearest neighborhood	NSDE [31]	Neighborhood mutation based speciation DE
7		NCDE [31]	Neighborhood mutation based crowding DE
8		LIPS [32]	Locally informed particle swarm optimizer
9		DE/nrand/1 [34]	DE with a neighborhood mutation strategy that uses the nearest neighbor as the base vector
10		dADE/nrand/1 [35]	DE/nrand/1 with an external archive and adaptive parameter selection strategy
11	Fuzzy neighborhood	Fast-LIPS [40]	Fast version of LIPS based on locality sensitive hashing
12		Fast-NCDE [40]	Fast version of NCDE based on locality sensitive hashing
13		PNPCDE [38]	Proximity-based crowding DE with parent-centric neighborhood mutation operator
14		LoICDE [39]	Locally informative crowding DE
15		LoISDE [39]	Locally informative speciation DE
16	Voronoi neighborhood	VNCDE	Voronoi neighborhood based crowding DE

V. EXPERIMENTAL STUDY

In this section, we carry out experiments to study the performance of the multimodal algorithms developed based on the Voronoi neighborhood. The effects of the algorithmic components are investigated in this section as well.

A. Experimental Setup

1) Algorithms in Comparison: We compare VNCDE with a number of state-of-the-art multimodal algorithms that are based on the index-based neighborhood, the nearest neighborhood, and the fuzzy neighborhood. In total, 16 algorithms are involved in the comparison. They are listed in Table II. All the algorithms are implemented in C++ and are compiled using the Microsoft Visual C++ compiler. They are executed on a computer with Intel Xeon X5675 CPU and 12-GB RAM. The correctness of the code has been verified through comparison with the results reported in the literature.

2) Test Functions: The benchmark function set for the CEC2013 special section on multimodal optimization [53] is adopted to test the performance of multimodal algorithms. The benchmark set contains 20 multimodal functions with various characteristics. The first ten functions F1–F10 are basic multimodal functions commonly used in the evolutionary multimodal optimization community. The remaining ten functions (F11–F20) are composite functions constructed by combining the basic functions. All the test functions are to be maximized. The mathematical descriptions of the test functions and their characteristics can be found in [53].

3) Parameter Settings: It is worth noting that throughout this paper, the term "parameter-free" is used to describe the Voronoi neighborhood. The algorithm built upon the Voronoi neighborhood (VNCDE) does not get rid of parameters. The parameters inherited from CDE are set as follows. The population size is fixed at 100. The scale factor F and the crossover probability Cr are set to 0.5 and 0.9, respectively. DE/inrand/1, DE/nrand/1, dADE/nrand/1, r2pso, r2pso-lhc, r3pso, and r3pso-lhc are algorithms developed without using niching parameters. For the other compared algorithms, their parameters are determined based on the recommendations of their developers and our empirical investigations. We have examined several different settings for each algorithm and have chosen the best performing one. For NCDE and NSDE, the neighborhood size parameter m is set to 10% and 20% of the population size, respectively. Similarly, in PNPCDE, the parameter k related to the neighborhood size is fixed at $10\%N$. In LoICDE and LoISDE, k is dynamically decreased from $12.5\%N$ to $5\%N$. In LIPS, *nsize* is dynamically increased from 5 to 8. The above parameter settings are mostly consistent with those suggested by the developers.

The termination criterion of the algorithms is defined by the maximum number of function evaluations (*MaxFEs*). The settings of *MaxFEs* for the 20 benchmark functions are kept the same as those suggested in [53]. For each test function, 50 independent runs are performed to obtain statistically reliable results.

4) Performance Metrics: To determine whether a peak has been located, an accuracy level ε needs to be specified. If the difference between the height of a peak and the fitness of an individual is less than ε, then the peak is considered to be found. It can be inferred that the higher the accuracy requirement, the more difficult the optimization task. In the experiment, a challenging accuracy level, i.e., ε = 1E-04, is employed. Given the accuracy level, the number of optimal solutions found by an algorithm can be computed via the specialized algorithm described in [53]. Two performance metrics are adopted to evaluate the performance of the multimodal algorithms.

a) Peak ratio: Peak ratio (PR) is the average percentage of global peaks found over multiple independent runs. It is calculated as follows:

$$\mathrm{PR} = \frac{\sum_{i=1}^{\mathrm{NR}} \mathrm{NPF}_i}{\mathrm{NPK} \cdot \mathrm{NR}} \tag{13}$$

where NPF_i is the number of peaks found in the ith run. NPK and NR in the denominator denote the number of global peaks and the number of runs, respectively.

b) Success rate: Success rate (SR) is the ratio of the number of successful runs (NSR) to the total number of runs (NR). A successful run of an algorithm is a run in which all the global peaks are located. It is computed according to the following formula:

$$\text{SR} = \frac{\text{NSR}}{\text{NR}}. \tag{14}$$

B. Overall Performance

The experimental results of the algorithms are reported in Table S1 in the supplementary material, where the PR and SR values are provided. Moreover, to determine the statistical difference between the PR values of VNCDE and those of the other algorithms, the Wilcoxon rank sum tests are conducted at significance level $\alpha = 0.05$. The notations "#" and "b" in the table indicate that the PR values achieved by VNCDE are significantly better and worse than that of its competitor, respectively. From the table, it can be observed that VNCDE outperforms the other algorithms on more than nine test functions. VNCDE achieves the 100% PR and 100% SR for the test functions F1–F5, F10, and F11, which suggests that it can effectively solve these problems. According to the reported PR values, VNCDE also consistently converges to multiple peaks for the complex composite problems with rugged landscape (F13–F20). When compared with NCDE, LoICDE, and Fast-NCDE, VNCDE exhibits better performance on 13, 11, and 9 functions, respectively. This reveals the advantage of the Voronoi neighborhood over the nearest neighborhood and the fuzzy neighborhood. The enhanced performance of VNCDE is largely attributed to the ESE and the multiple-strategy search mechanism, which make systematic use of the structure information encoded in the Voronoi neighborhoods of individuals.

To further investigate the differences between VNCDE and the competitors, the multiple-problem Wilcoxon's test and the Friedman's test are conducted by using the KEEL software [54]. Table S2 in the supplementary material provides the results of the Wilcoxon's test. We can observe that VNCDE has higher $R+$ values than $R-$ values in all the cases and the corresponding p values are less than 0.05, indicating that VNCDE is superior to the competitors on the 20 multimodal functions. The overall rankings of the algorithms provided by the statistical test are summarized in Table III. As shown in the table, VNCDE has the highest ranking, followed by algorithms that are based on the fuzzy neighborhood and the nearest neighborhood. The performance of the algorithms with the index-based neighborhood structure is slightly worse than those in the other categories.

Moreover, we have studied the convergence speed of the algorithms and the efficacy of the multiple-strategy search mechanism. The details are provided in the supplementary material due to the space limit.

TABLE III
RANKINGS OF THE ALGORITHMS BY THE FRIEDMAN'S TEST

Algorithm	Ranking	Algorithm	Ranking
VNCDE	3.15	PNPCDE	8
Fast-NCDE	4.475	LIPS	8.1
Fast-LIPS	5.1	LoISDE	9.525
dADE/nrand/1	6.225	NSDE	11.1
DE/nrand/1	7.75	r3pso-lhc	11.3
DE/inrand/1	7.775	r2pso-lhc	11.75
LoICDE	7.925	r3pso	12.525
NCDE	8	r2pso	13.3

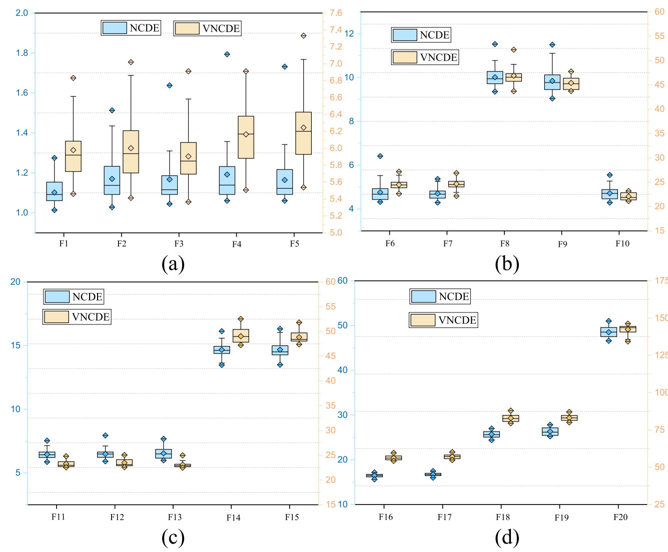


Fig. 7. Running time comparison of NCDE and VNCDE. (a) F1–F5. (b) F6–F10. (c) F11–F15. (d) F16–F20.

C. Running Time

In this section, we compare the running time of the algorithms. The experimental results are provided in Table S3 in the supplementary material. Moreover, Fig. 7 visualizes the running time of NCDE and VNCDE with box plots. The results are averaged over 50 independent runs and the time used for fitness evaluations has been counted as well. All the algorithms are executed in the same environment.

From the table, it can be observed that the index-based algorithms (DE/inrand/1 and rpso) are the fastest among all the compared algorithms. For the basic multimodal problems (F1–F10), the running time is dominated by the primitive operations of the algorithms. For complex composite problems (F11–F20), the running time is dominated by the evaluation of the objective functions. Fast-NCDE and Fast-LIPS are able to achieve similar time efficiency as the index-based algorithms. This is attributed to the use of locality sensitive hash functions, which alleviate the need of distance calculations. The nearest and fuzzy neighborhood-based algorithms generally spend more time than index-based algorithms. As indicated by the complexity analysis, VNCDE is the most time-consuming algorithm. The time spent by VNCDE is about three times longer than NCDE, as shown in Fig. 7.

The high complexity of VNCDE mainly comes from the construction of Voronoi neighborhoods. Algorithm 1 is designed for the purpose of illustrating the usefulness of Voronoi neighborhoods. It is very simple and primitive. We believe that more advanced algorithms can be developed to achieve quadratic or even linear time complexity through extended future research.

D. Comparison With Four Recent Algorithms

In this section, we compare VNCDE with four more recently proposed algorithms, i.e., MOMMOP [43], LMCEDA [36], LMSEDA [36], and EMO-MMO [55]. MOMMOP is a novel algorithm that transforms a multimodal problem into a multiobjective problem. The transformed problem has mutually conflicting objectives and each Pareto optimal solution corresponds to a peak of the original problem. LMCEDA and LMSEDA are two algorithms based on clustering and estimation of distribution. Three strategies, namely, a dynamic cluster sizing strategy, a reproduction strategy that alternatively utilizes Gaussian and Cauchy distributions, and an adaptive local search strategy, have been incorporated in the two algorithms to enhance the niching performance. EMO-MMO divides the optimization process into three stages. In the first stage, an approximate fitness landscape is obtained by solving the transformed multiobjective problem. In the second stage, a peak detection method is used to find regions where optimal solutions may exist. In the third stage, local search is performed to find the optimal solutions inside the detected peak regions.

The experimental results at accuracy level ε = 1E-04 are presented in Table S4 in the supplementary material, where the highest PR values are marked in bold. The results of MOMMOP, LMCEDA, LMSEDA, and EMO-MMO are taken directly from their corresponding publications. Table S5 in the supplementary material lists the rankings of the algorithms provided by the Friedman's test.

As can be seen in Table S4 in the supplementary material, MOMMOP performs extremely well on the first ten problems. All the global optima are successfully located by the algorithm. EMO-MMO yields the highest PR and SR values on test problems F1–F8 and F10–F14. However, their performance deteriorates when solving composite problems with a larger number of decision variables. In MOMMOP, each decision variable is used to design two objectives. The number of objectives increases linearly with the number of variables. This leads to the loss of selection pressure for the nondominated sorting-based operator. For EMO-MMO, approximating the entire multimodal landscape becomes very difficult since the size of the search space grows exponentially with the dimensionality. In comparison, VNCDE and LMSEDA have relatively good performance on the composition problems (F15–F20). Equipped with the Voronoi neighborhood-based search strategies, VNCDE is capable of producing competitive results on problems with different numbers of dimensions. According to the Friedman's test given in Table S5 in the supplementary material, VNCDE has the overall best performance. EMO-MMO and MOMMOP are in the second and third place, respectively.

E. Effect of the Evolutionary State Estimator

In this section, we proceed to study the effect of the ESE. Fig. 8 plots the three groups of individuals identified by the ESE at the 20th iteration of VNCDE when solving F6 and F10. In the figure, the *dominators*, *challengers*, and *explorers* are represented by stars, circles, and triangles, respectively. According to the figures, it can be seen that the evolutionary states of the individuals are correctly identified. The *dominators* are the most likely individuals to reach the peaks and they are generally surrounded by the *challengers*. Some of the *dominators* are near local optima [Fig. 8(a)]. In contrast, the *explorers* are confined in valleys between the peaks or in outer ranges of subpopulations [Fig. 8(b)]. These observations confirm the efficacy of the approximation algorithm in finding Voronoi neighbors and the validity of the ESE in classifying the individuals.

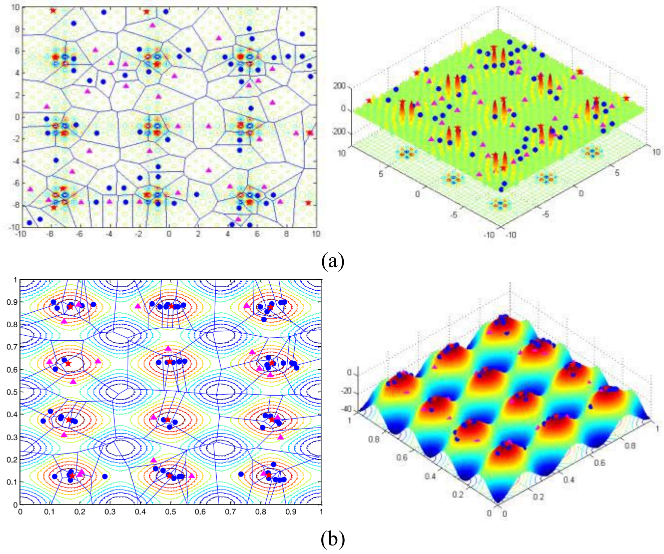


Fig. 8. Three groups of individuals identified by the ESE on 2-D multimodal problems. (a) F6. (b) F10.

F. Effect of Voronoi Neighborhood-Based Search Mechanism

To assess the impact of Voronoi neighborhood-based search mechanism and crowding on the performance of VNCDE, we compare the proposed algorithm with VNDE and CDE. VNDE is a simplified variant of VNCDE that abandons the crowding-based selection mechanism. The experimental results of the three algorithms at accuracy level ε = 1E-04 are listed in Table S6 in the supplementary material. The highest PR values are highlighted in bold. We can observe from the table that VNCDE achieves the best results on most of the test problems. According to the comparison results summarized in the last row the table, VNCDE outperforms VNDE on 15 out of 20 test problems, and outperforms CDE on 12 test problems. The PR values obtained by VNDE and CDE reveal that both the Voronoi neighborhood-based search mechanism and the crowding technique are effective in inducing multiple convergence behavior. More importantly, combing the two mechanisms endows VNCDE with the ability to reach a higher level of performance.

The advantages of Voronoi neighborhood are twofold. First, it is parameter free, so the algorithm built upon it will not be influenced by the problem-related niching parameters. Second, it provides useful information about the spatial distribution of individuals, which can be exploited to reveal the evolutionary states. The evolutionary states can subsequently be used to guide the selection of suitable reproduction strategies, so the individuals are capable of exploiting their neighborhoods more effectively.

To further demonstrate the advantages, we compare four variants of VNCDE that are based on different types of neighborhoods. The variants are denoted by IDE, NDE, VNDE-random, and VNDE, respectively. In IDE, NDE, and VNDE-random, an offspring solution is produced by randomly drawing a strategy from the strategy pool containing Gaussian local search, DE/current-to-*nbest*, and DE/rand/1. The only difference between the algorithms is that the candidate vectors ($\boldsymbol{x}_{r1}$, $\boldsymbol{x}_{r2}$, and $\boldsymbol{x}_{r3}$) are selected from index-based neighborhood, k-nearest neighborhood, and Voronoi neighborhood. In VNDE, the search strategy is chosen based on the role of the parent identified by ESE. In order to isolate the effect of the three types of neighborhoods, the traditional one-to-one selection mechanism of DE is adopted instead of the crowding technique. Table S7 in the supplementary material provides the PR and SR values obtained by the four algorithms. The best PR values are marked in bold. The second and third last rows summarize the win/tie/loss counts for VNDE-random and VNDE, while the last row gives the rankings of the algorithms obtained by the Friedman's test.

It can be observed from Table S7 in the supplementary material that VNDE-random outperforms IDE and NDE on majority of the test problems. The index-based method often suffers from the oscillation problem due to the ignorance of the spatial correlation of individuals. The k-nearest neighborhood needs to specify the size parameter and it does not take into account the structural information. This may bias the search toward a specific direction. In comparison, the Voronoi neighborhood gathers all the surrounding individuals to facilitate the search of peaks in local environments. The comparison result of IDE, NDE, and VNDE-random confirms the first advantage of the Voronoi neighborhood.

Another observation can be made is that by resorting to the information extracted by ESE, VNDE succeeded in improving the performance of VNDE-random on ten test problems. The evolutionary states of individuals revealed by the Voronoi neighborhood and ESE are helpful in determining the proper use of the search strategies, which contributes to the enhancement of the multiple convergence ability. The comparison result of VNDE-random and VNDE validates the second advantage of the Voronoi neighborhood.

G. Effect of Problem Dimensionality

To investigate the effect of problem dimensionality on the performance of VNCDE, we carry out experiments on three scalable test problems, i.e., the modified Rastrigin problem and two composition problems (CF1 and CF2). The dimensionality of the problems ranges from 20 to 50. For the modified Rastrigin problem, the termination criterion is set as $MaxFEs = 5000D$. For CF1 and CF2, the termination criterion is set as $MaxFEs = 10\,000D$. Six competitive algorithms are involved in the comparison, namely, DE/inrand/1, NCDE, dADE/nrand/1, LoICDE, Fast-NCDE, and VNCDE. Their PR values are summarized in Table S8 in the supplementary material. The highest PR value for each test problem is emphasized in bold. The running time of the algorithms (in seconds) is reported in the table as well.

From the table, it can be seen that the VNCDE yields the best results on 13 problem instances and Fast-NCDE yields the best results 11 problem instances. For the modified Rastrigin problem, the performance of the compared algorithms remains promising under different numbers of dimensions. However, when solving the complex composite problems CF1 and CF2, the PR values obtained by the algorithms drop below 0.5. Moreover, the performance deteriorates as the number of dimensions increases. As for the computational complexity, the index-based algorithm DE/inrand/1 consumes the least time. The running time of Fast-NCDE is very close to DE/inrand/1 owing to the use of efficient LSH functions. In comparison, the time spent by VNCDE increases more rapidly than the other algorithms. Fig. S1 in the supplementary material depicts the growth trend of the running time for the modified Rastrigin problem and CF1. As shown in the figure, the computational complexities of the algorithms are linear in the number of variables.

H. Effect of Population Size

To study the effect of population size on the performance VNCDE, experiments are conducted on a problem with a large number of global optima (the 3-D Vincent problem with 216 global optima) and the composite problems using ten different settings of population size. Specifically, the population size ranges from 20 to 200 with an increment of 20. The other settings of the experiment remain unchanged. The experimental results are tabulated in Table S9 in the supplementary material, where the PR values and the running time are provided. Fig. S2 in the supplementary material plots the changes of the PR values along with the growth of the population size, while Fig. S3 in the supplementary material displays the outcome of the running time.

From the table and the figures, we can make the following observations. For the Vincent problem, the PR values of the algorithms generally increase with the population size. The reason is that the number of global optima is larger than the number of individuals, increasing the population size helps to cover more promising search regions and increase the number of located peaks. For the composition problems CF1 and CF2, the PR values of the algorithms are less susceptible to the setting of the population size. The performance of VNCDE is more stable than the other algorithms. It achieves the highest PR values on CF1 and CF2 under different population sizes. The robustness of VNCDE is mainly attributed to the use of multiple search strategies, which enhances the search flexibility of individuals in various scenarios. As for the computational complexity, it can be observed that the running time of DE/inrand/1 and Fast-NCDE is not influenced by the population size. In comparison, the pairwise distance calculations for the population members prolong the running time of NCDE and LoICDE. As revealed by the complexity analysis, the time spent by VNCDE increases more significantly as the population size grows.

VI. CONCLUSION

In this paper, a new neighborhood concept is introduced based on a geometrical construction called Voronoi diagram.

We showed the distinct properties of the Voronoi neighborhood by comparing it with three existing concepts, namely, the index-based neighborhood, the nearest neighborhood, and the fuzzy neighborhood. Different from the existing concepts, the Voronoi neighborhood is parameter-free and it contains more information about the spatial distribution of individuals. To ensure the applicability of the concept, a polynomial time approximation algorithm for finding Voronoi neighbors in high dimensional search space is developed. Then, the spatial information encoded in the Voronoi neighborhood is extracted via an ESE. The ESE categorizes the individuals in the population into three different groups, i.e., *dominators*, *challengers*, and *explorers*. Each group of individuals are treated differently according to their characteristics. Finally, a new niching algorithm termed VNCDE is presented by combining the Voronoi neighborhood-based search mechanism with the crowding technique.

We have conducted a number of experiments to assess the performance VNCDE. The numerical results indicate that the ESE is able to correctly estimate the evolutionary states of the individuals. Moreover, the Voronoi neighborhood-based search strategies tailored for the *dominators*, *challengers*, and *explorers* can greatly enhance their search efficiency. With the new algorithmic components, VNCDE succeeded in locating more optimal solutions than several state-of-the-art algorithms. One major drawback of VNCDE is that it consumes much more time than the compared algorithms. This is due to the heavy computational burden incurred by Voronoi neighbor finding.

In the future, it would be beneficial to develop more efficient approximation algorithms to reduce the time complexity of finding Voronoi neighbors, as well as to reduce the number of Type II errors. Another plan is to test the performance of VNCDE on more challenging benchmark problems [56] and some real-world multimodal problems. To avoid VNCDE overfitting to the benchmark problems and to increase its generalizability to real-world applications, it is desirable to employ automatic parameter tuning tools like irace [57]. Finally, it remains to be investigated whether the Voronoi neighborhood can be used to facilitate the design of genetic operators for dynamic optimization and multiobjective optimization.

REFERENCES

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

[2] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.

[3] H. G. Beyer and H. P. Schwefel, "Evolution strategies—A comprehensive introduction," *Nat. Comput.*, vol. 1, no. 1, pp. 3–52, 2002.

[4] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proc. Congr. Evol. Comput. (CEC)*, 1999, pp. 1945–1950.

[5] J. Kennedy, "Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance," in *Proc. Congr. Evol. Comput. (CEC)*, 1999, pp. 1931–1938.

[6] M. Nasir, S. Das, D. Maity, S. Sengupta, U. Halder, and P. N. Suganthan, "A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization," *Inf. Sci.*, vol. 209, pp. 16–36, Nov. 2012.

[7] P. N. Suganthan, "Particle swarm optimiser with neighborhood operator," in *Proc. Congr. Evol. Comput. (CEC)*, 1999, pp. 1958–1962.

[8] T. Peram, K. Veeramachaneni, and C. K. Mohan, "Fitness-distance-ratio based particle swarm optimization," in *Proc. IEEE Swarm Intell. Symp.*, 2003, pp. 174–181.

[9] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: Simpler, maybe better," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 204–210, Jun. 2004.

[10] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential evolution using a neighborhood-based mutation operator," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 526–553, Jun. 2009.

[11] M. G. Epitropakis, D. K. Tasoulis, N. G. Pavlidis, V. P. Plagianakos, and M. N. Vrahatis, "Enhancing differential evolution utilizing proximity-based mutation operators," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 99–119, Feb. 2011.

[12] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[13] S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Decomposition-based multiobjective evolutionary algorithm with an ensemble of neighborhood sizes," *IEEE Trans. Evol. Comput.*, vol. 16, no. 3, pp. 442–446, Jun. 2012.

[14] Z. Yu, H. S. Wong, D. Wang, and M. Wei, "Neighborhood knowledge-based evolutionary algorithm for multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 15, no. 6, pp. 812–831, Dec. 2011.

[15] R. Storn and K. Price, "Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces," Int. Comput. Sci. Inst., Berkeley, CA, USA, Rep. TR-95–012, 1995.

[16] S. Das, S. S. Mullick, and P. N. Suganthan, "Recent advances in differential evolution—An updated survey," *Swarm Evol. Comput.*, vol. 27, pp. 1–30, Apr. 2016.

[17] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," in *Proc. 2nd Int. Conf. Genet. Algorithms*, 1987, pp. 41–49.

[18] R. Thomsen, "Multimodal optimization using crowding-based differential evolution," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2004, pp. 1382–1389.

[19] A. Petrowski, "A clearing procedure as a Niching method for genetic algorithms," in *Proc. IEEE Int. Conf. Evol. Comput.*, May 1996, pp. 798–803.

[20] J.-P. Li, M. E. Balazs, G. T. Parks, and P. J. Clarkson, "A species conserving genetic algorithm for multimodal function optimization," *Evol. Comput.*, vol. 10, no. 3, pp. 207–234, Sep. 2002.

[21] J. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI, USA: Univ. Michigan Press, 1975.

[22] K. D. Jong, "An analysis of the behavior of a class of genetic adaptive systems," Ph.D. dissertation, Dept. Comput. Commun. Sci., Univ. Michigan, Ann Arbor, MI, USA, 1975.

[23] S. W. Mahfoud, "Niching methods for genetic algorithms," Ph.D. dissertation, Dept. Comput. Sci., Univ. Illinois Urbana–Champaign, Champaign, IL, USA, 1995.

[24] D. Parrott and X. Li, "Locating and tracking multiple dynamic optima by a particle swarm model using speciation," *IEEE Trans. Evol. Comput.*, vol. 10, no. 4, pp. 440–458, Aug. 2006.

[25] X. Li, "Efficient differential evolution using speciation for multimodal function optimization," in *Proc. Conf. Genet. Evol. Comput.*, 2005, pp. 873–880.

[26] N. Lynn and P. N. Suganthan, "Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation," *Swarm Evol. Comput.*, vol. 24, pp. 11–24, Oct. 2015.

[27] S. Hui and P. N. Suganthan, "Ensemble and arithmetic recombination-based speciation differential evolution for multimodal optimization," *IEEE Trans. Cybern.*, vol. 46, no. 1, pp. 64–74, Jan. 2016.

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

[29] M. G. Epitropakis, V. P. Plagianakos, and M. N. Vrahatis, "Multimodal optimization using Niching differential evolution with index-based neighborhoods," in *Proc. IEEE Int. Conf. Evol. Comput.*, 2012, pp. 1–8.

[30] X. D. Li, "A multimodal particle swarm optimizer based on fitness Euclidean-distance ration," in *Proc. Genet. Evol. Comput. Conf.*, 2007, pp. 78–85.

[31] 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, Oct. 2012.

[32] 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, Jun. 2013.

[33] 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, Aug. 2014.

[34] M. G. Epitropakis, V. P. Plagianakos, and M. N. Vrahatis, "Finding multiple global optima exploiting differential evolutions Niching capability," in *Proc. IEEE Symp. Differ. Evol. (SDE)*, Apr. 2011, pp. 1–8.

[35] M. G. Epitropakis, X. Li, and E. K. Burke, "A dynamic archive Niching differential evolution algorithm for multimodal optimization," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2013, pp. 79–86.

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

[37] Q. Yang *et al.*, "Adaptive multimodal continuous ant colony optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 191–205, Mar. 2017.

[38] 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, Oct. 2014.

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

[40] Y.-H. Zhang, Y.-J. Gong, H.-X. Zhang, T.-L. Gu, and J. Zhang, "Toward fast Niching evolutionary algorithms: A locality sensitive hashing-based approach," *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 347–362, Jun. 2017.

[41] 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, Feb. 2010.

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

[43] 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, Apr. 2015.

[44] S. Das, S. Maity, B.-Y. Qu, and P. N. Suganthan, "Real-parameter evolutionary multimodal optimization: A survey of the state-of-the-art," *Swarm Evol. Comput.*, vol. 1, no. 2, pp. 71–88, Jun. 2011.

[45] 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, Aug. 2017.

[46] J. R. Sack and J. Urrutia, *Handbook of Computational Geometry*. Amsterdam, The Netherlands: Elsevier, 1999.

[47] M. L. Shamos, "Computational geometry," Ph.D. dissertation, Dept. Comput. Sci., Yale Univ., New Haven, CT, USA, 1978.

[48] M. I. Shamos and D. Hoey, "Closest-point problems," in *Proc. 16th Annu. Symp. Found. Comput. Sci.*, 1975, pp. 151–162.

[49] S. Fortune, "A sweepline algorithm for Voronoi diagrams," *Algorithmica*, vol. 2, nos. 1–4, p. 153, 1987.

[50] A. Bowyer, "Computing Dirichlet tessellations," *Comput. J.*, vol. 24, no. 2, pp. 162–166, 1981.

[51] D. F. Watson, "Computing the *n*-dimensional Delaunay tessellation with application to Voronoi polytopes," *Comput. J.*, vol. 24, no. 2, pp. 167–172, 1981.

[52] K. Q. Brown, "Voronoi diagrams from convex hulls," *Inf. Process. Lett.*, vol. 9, no. 5, pp. 223–228, 1979.

[53] X. Li, A. Engelbrecht, and M. G. Epitropakis, "Benchmark functions for CEC'2013 special session and competition on Niching methods for multimodal function optimization," Evol. Comput. Mach. Learn. Group, RMIT Univ., Melbourne, VIC, Australia, Rep., 2013.

[54] J. Alcalá-Fdez *et al.*, "KEEL: A software tool to assess evolutionary algorithms to data mining problems," *Soft Comput.*, vol. 13, no. 3, pp. 307–318, 2009.

[55] 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, Oct. 2018.

[56] B. Y. Qu, J. J. Liang, Z. Y. Wang, Q. Chen, and P. N. Suganthan, "Novel benchmark functions for continuous multimodal optimization with comparative results," *Swarm Evol. Comput.*, vol. 26, pp. 23–34, Feb. 2016.

[57] M. López-Ibáñez, J. Dubois-Lacoste, L. P. Cáceres, M. Birattari, and T. Stützle, "The irace package: Iterated racing for automatic algorithm configuration," *Oper. Res. Perspectives*, vol. 3, pp. 43–58, Sep. 2016.

Yu-Hui Zhang received the B.S. and Ph.D. degrees in computer science from Sun Yat-sen University, Guangzhou, China, in 2013 and 2019, respectively.

He is a Lecturer with the School of Computer Science and Technology, Dongguan University of Technology, Dongguan, China. His current research interests include evolutionary computation, swarm intelligence, and their real-world applications.

Yue-Jiao Gong (S'10–M'15) received the B.S. and Ph.D. degrees in computer science from Sun Yat-sen University, Guangzhou, China, in 2010 and 2014, respectively.

She is currently a Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou. Her current research interests include evolutionary computation, swarm intelligence, and their applications to intelligent transportation and smart city scheduling. She has published over 70 papers, including over 30 IEEE TRANSACTIONS papers, in the above areas.

Ying Gao received the bachelor's and master's degrees in computer science from the Central South University, Changsha, China, in 1997 and 2000, respectively, and the Ph.D. degree in computer science from the South China University of Technology, Guangzhou, China, in 2006.

She is currently a Professor with the School of Computer Science and Engineering, South China University of Technology. She has published over 30 papers in international journals and conferences. Her current research interests include service-oriented computing technology, software architecture, and network security.

Hua Wang received the Ph.D. degree from the University of Southern Queensland, Toowoomba, QLD, Australia.

He was a Professor with the University of Southern Queensland. He is currently a Full Time Professor with Victoria University, Melbourne, VIC, Australia. He has over ten years teaching and working experience in applied informatics at both enterprise and university. He has expertise in electronic commerce, business process modeling, and enterprise architecture. As a Chief Investigator, three Australian Research Council Discovery grants have been awarded since 2006, and 200 peer reviewed scholar papers have been published.

Jun Zhang (M'02–SM'08–F'17) received the Ph.D. degree from the City University of Hong Kong, Hong Kong, in 2002.

He is currently a Visiting Scholar with the University of Victoria, Melbourne, VIC, Australia. His current research interests include computational intelligence, cloud computing, data mining, and power electronic circuits. He has published over 200 technical papers in the above areas.

Dr. Zhang was a recipient of the China National Funds for Distinguished Young Scientists from the National Natural Science Foundation of China in 2011 and the First-Grade Award in Natural Science Research from the Ministry of Education, China, in 2009. He is currently an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the IEEE TRANSACTIONS ON CYBERNETICS, and the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS.