

Article

# A Competitive Memory Paradigm for Multimodal Optimization Driven by Clustering and Chaos

Jorge Gálvez <sup>1,\*</sup> , Erik Cuevas <sup>1</sup>  and Krishna Gopal Dhal <sup>2</sup>

<sup>1</sup> Departamento de Electrónica, Universidad de Guadalajara, CUCEI, Av. Revolución 1500, Guadalajara C.P. 44430, Jal, Mexico; erik.cuevas@ucei.udg.mx

<sup>2</sup> Department of Computer Science and Application, Midnapore College (Autonomous), Paschim Medinipur, West Bengal 721101, India; krishnagopal.dhal@midnaporecollege.ac.in

\* Correspondence: jorge.galvez@academicos.udg.mx; Tel.: +52-33-1378-5900 (ext. 27714)

Received: 29 April 2020; Accepted: 3 June 2020; Published: 8 June 2020



**Abstract:** Evolutionary Computation Methods (ECMs) are proposed as stochastic search methods to solve complex optimization problems where classical optimization methods are not suitable. Most of the proposed ECMs aim to find the global optimum for a given function. However, from a practical point of view, in engineering, finding the global optimum may not always be useful, since it may represent solutions that are not physically, mechanically or even structurally realizable. Commonly, the evolutionary operators of ECMs are not designed to efficiently register multiple optima by executing them a single run. Under such circumstances, there is a need to incorporate certain mechanisms to allow ECMs to maintain and register multiple optima at each generation executed in a single run. On the other hand, the concept of dominance found in animal behavior indicates the level of social interaction among two animals in terms of aggressiveness. Such aggressiveness keeps two or more individuals as distant as possible from one another, where the most dominant individual prevails as the other withdraws. In this paper, the concept of dominance is computationally abstracted in terms of a data structure called “competitive memory” to incorporate multimodal capabilities into the evolutionary operators of the recently proposed Cluster-Chaotic-Optimization (CCO). Under CCO, the competitive memory is implemented as a memory mechanism to efficiently register and maintain all possible optimal values within a single execution of the algorithm. The performance of the proposed method is numerically compared against several multimodal schemes over a set of benchmark functions. The experimental study suggests that the proposed approach outperforms its competitors in terms of robustness, quality, and precision.

**Keywords:** evolutionary algorithms; multimodal optimization; competitive memory; collective behavior; Cluster-Chaotic-Optimization

---

## 1. Introduction

Engineering optimization aims to obtain the optimal solution from a possible set of candidate solutions for a given minimization/maximization problem [1,2]. Many areas, such as economics, science, bio-engineering, and others, model an optimization problem in terms of objective functions. Traditionally, to solve optimization problems, engineering optimization has proposed the use of classical deterministic paradigms which theoretically guarantee the location of global optima. However, classical approaches present issues in the presence of multiple optima [3,4]. Deterministic methods are susceptible to being trapped in local optima. Under such circumstances, these methodologies obtain suboptimal values. On the other hand, Evolutionary Computation Methods (ECMs) have been proposed to solve complex optimization problems to alleviate the stagnation problem derived from the presence of multiple optima in a given objective function. ECMs are catalogued as stochastic search

mechanisms, where the use of evolutionary operators guides the search strategy towards the global optimum [5,6]. To this end, many scientific, engineering, and even economic research communities have adopted the use of ECMs as a generic tool to solve optimization problems, regardless of the real constraints found in their mathematical models.

Recently, several ECMs have been proposed by using biological, natural, or even social metaphors as search strategies to improve the detection of optimal values, regardless of the domain, nonconvexity, and complexity of the given objective functions. ECMs are developed as the synergy among randomness and deterministic criteria to imitate the behavior of their abstracted metaphors. Under ECMs, the optimization process is divided into two major parts: exploration and exploitation. The exploration stage aims to make each search agent as scattered as possible in the entire search space, while in the exploitation stage, each search agent is disturbed, causing it to search in more promising areas. These two evolutionary stages are abstracted from many metaphors found in nature, i.e., biological or even social phenomena. Under such an assumption, the evolutionary operators of each ECM are computationally implemented based on the abstracted knowledge of their behavior.

Some ECMs consider biological aspects of the genetic recombination of parents to produce fitter individuals. That is the idea behind the Genetic Algorithm (GA), which has been proposed by Holland [7] and Goldberg [8]. Other methods apply the concept of the collective intelligence of swarms for finding food. For example, the Ant Colony Optimization algorithm (ACO) was proposed by Dorigo [9], the Particle Swarm Optimization was proposed by Kennedy and Eberhart [10], and the Artificial Bee Colony Optimization algorithm (ABC) was proposed by Karaboga [11]. On the other hand, several ECMs have been proposed following mathematical or physical principles. Under this scenario, Rashedi [12] considers the gravitational force among bodies to generate an evolutionary operator in the Gravitational Search Algorithm (GSA). Also, Storn and Price [13] proposed the weighted difference between two vectors to form a simplex in the Differential Evolution (DE) method. Hansen [14] uses the covariance among search agents to guide the search strategy in the Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES).

The main objective of most of the proposed ECMs is to find the optimal value for a given objective function. However, the global solution may be expensive, impractical, or physically unrealizable. Under such limitations, multimodal strategies are required in order to efficiently detect and store multiple solutions. In multimodal optimization problems, the aim of locating all possible optima consists of obtaining a set of optimal and suboptimal solutions. This obtained set will contain many solutions, where each considers the realistic constraints found in the mathematical description of a given problem. Then, the user must select which solution will be used based on his/her expertise. To achieve multiple local/global optima detections, a multimodal search strategy must be designed considering at least three aspects: (i) a storage structure to maintain all possible optima; (ii) a mechanism to add/remove elements in the storage unit; and (iii) an update mechanism for managing the storage structure to avoid redundancies.

Under such circumstances, many multimodal approaches have been developed to incorporate multimodal capabilities on single optimum ECMs. In the multimodal literature, multimodal techniques can be classified into two major groups: crowding-niching and speciation [15]. In crowding-niching approaches, the population is divided into niches of different species, in contrast to speciation methods, where the population is divided into individuals of the same species. Based on crowding-niching approaches, some multimodal methods have been proposed.

Based on crowding-niching techniques, Sareni proposed the Fitness Sharing (FS) method [16] as a mechanism to incorporate a multimodal strategy in the optimization process. This technique considers two aspects: a similarity function and a sharing model. The sharing model is based on affinity among individuals, while the similarity function uses the distance among individuals to calculate such affinities. The resulting scheme assigns a decreasing fitness value to similar individuals to reduce the number of redundancies in the solutions. Some ECMs have been proposed as multimodal approaches

based on the FS technique: Fitness Sharing Differential Evolution [17], Fitness Euclidean-distance Differential Evolution [18], and Information Sharing Artificial Bee Colony [19].

Speciation techniques divide the population in terms of species. That is, they group individuals of the same species. Additionally, speciation methods can be classified into topology- and distance-based approaches. In distance-based mechanisms, the similarity of individuals is computed considering spatial relationships. Under this category, the Differential Evolution with Self-adaptive Strategy [20] and the Elitist-population Genetic Algorithm [21] have been proposed. In topology mechanisms, the relationship between location and fitness value are considered. In this category, the History-based Topological Speciation [22] method, Recursive Middling [23], and the hill-valley approach [24] have been proposed.

The previously described multimodal approaches have been presented as alternative methodologies to detect and efficiently store all possible optimal values by considering single objective functions. However, some researchers have proposed techniques based on the principles of multiobjective optimization methods [25–28], that is, by controlling one or more conflictive objectives such as diversification and intensification. Under this paradigm, the Multiobjective Optimization for Multiple optimal of Multimodal Optimization algorithm [28], the Bi-objective Multipopulation Genetic Algorithm [25], and the Multimodal Optimization with Bi-objective Differential Evolution [27] methods have been proposed.

On the other hand, the Cluster-Chaotic-Optimization (CCO) [29] was recently proposed, integrating clustering and chaotic sequences to find the optimal values of optimization problems. The CCO method divides the population into clusters considering intra- and extra- cluster operations. In intracuster procedures, the search mechanism locally explores each formed cluster. In extracuster procedures, the search mechanism globally explores the entire search space for a given optimization problem.

In the beginning, CCO considers each data point (individual) as a single cluster. Then, during the optimization process, each data point is grouped with similar data points. The way CCO groups data points is based on the hierarchical clustering method [30], which considers the capacity to generate associations (links) among data points based on their variance, regardless of the a priori specification of the number of clusters. During the grouping process, the stochastic characteristics of chaos theory are also implemented in CCO. In CCO, chaotic sequences have been adopted in order to produce randomness. This procedure increases the performance and the population diversity of the CCO. As a consequence, the original CCO operators present a balance among the exploration and exploitation stages.

Considering the inter- and extra- cluster stages of the CCO, the evolutionary operators naturally decompose the population into clusters; this process can detect potential optima in a single run. Under such circumstances, CCO can be adapted to incorporate multimodal capabilities.

This paper presents a novel approach for the detection of multiple optima in any optimization problem. The proposed method, called Multimodal Cluster-Chaotic-Optimization (MCCO), uses the original structure of CCO operators and extends its functionality to efficiently detect, register, and maintain multiple optima for a given objective function. Under the proposed approach, a computational data structure called the “competitive memory mechanism” is adopted as an abstraction of the concept of dominance found in animal interactions [31–33]. The performance of MCCO was numerically compared against some state-of-the-art multimodal techniques over a set of multimodal benchmark functions. The experimental results were statistically validated by a nonparametric test. The experimental results suggest that MCCO outperformed other multimodal approaches in terms of accuracy, robustness, and consistency in most of the benchmark functions.

The rest of the paper is organized as follows. In Section 2, the CCO method is described. In Section 3 the proposed MCCO is presented. Section 4 describes the performance of the proposed MCCO against several multimodal approaches, considering a set of multimodal benchmark functions. Finally, in Section 5, conclusions are presented.

## 2. Cluster-Chaotic-Optimization (CCO)

The main process of the CCO is based on data analysis of the population through a clustering method. The method considered here is the Ward method. With this approach, individual associations among data points guide the search strategy for the optimization process. Traditionally, most of the proposed ECMs consider each individual separately, regardless of the spatial information among them. Under this situation, CCO considers spatial associations among each generation to group similar individuals into clusters. These clusters will operate locally and globally to improve the search strategy. On the other hand, CCO also incorporates the stochastic behavior of chaotic sequences to randomly perturb solutions. The use of chaotic sequences has been demonstrated to improve the performance of ECMs based on random numbers [34–36].

The CCO method was conceived to find the global optimum of complex optimization problems in the following form:

$$\begin{aligned} &\text{minimize/maximize } J(\mathbf{x}) \quad \mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n \\ &\text{subject to} \quad \mathbf{x} \in \mathbf{X} \end{aligned} \tag{1}$$

where  $J: \mathbb{R}^n \rightarrow \mathbb{R}$  corresponds to the objective function, and  $\mathbf{X} = \{ \mathbf{x} \in \mathbb{R}^n | l_j \leq x_j \leq u_j, j = 1, \dots, n \}$  is a bounded searched space, constrained by the upper ( $u_j$ ) and lower ( $l_j$ ) bounds. To find the global optimum of the aforementioned definition, CCO considers population  $\mathbf{D}^k (\{ \mathbf{d}_1^k, \mathbf{d}_2^k, \dots, \mathbf{d}_{N_D}^k \})$  compound of  $N_D$  data points which evolves from an initial iteration ( $k = 0$ ) to a maximum number of *gen* iterations ( $k = \text{gen}$ ). Each solution  $\mathbf{d}_i^k$  ( $i \in [1, \dots, N_D]$ ) corresponds to a  $n$ -dimensional vector  $\{ d_{i,1}^k, d_{i,2}^k, \dots, d_{i,n}^k \}$  where each dimension represents a decision variable.

Based on the population description in CCO, three procedures are required to implement the evolution process of each data point: the first corresponds to the initialization process of the population of data points; the second considers an intracluster operation to search locally inside each cluster; and finally, an extraccluster operation is executed to globally search outside each cluster but inside the search space.

### 2.1. Initialization

CCO begins by randomly initializing the population  $\mathbf{D}^k$  compound of  $N_D$  solutions (data points). Each dimension of every data point corresponds to a uniform random number within the range of the upper ( $u_j$ ) and lower ( $l_j$ ) bounds. This mechanism is mathematically defined as follows:

$$d_{i,j}^k = l_j + \text{rand}(0,1) \cdot (u_j - l_j), \quad j = 1, 2, \dots, n, \quad i = 1, 2, \dots, N_D \tag{2}$$

where  $d_{i,j}^k$  represents the  $j$ -th decision variable of the  $i$ -th solution (data point) at the  $k$  iteration.

### 2.2. Intracluster Operation

In CCO, the main process for identifying promissory search zones within the search space is conducted by clustering. During the optimization process of CCO, clustering starts grouping individuals into a hierarchical structure. This mechanism generalizes natural data associations without considering the total number of clusters.

In each iteration of the CCO algorithm, the Ward method is used to obtain clusters by spatial data associations. Then, an evolutionary operator called “Intracluster operation” will locally explore each formed cluster. Under this operation, two procedures are computed: the local attraction mechanism and the local perturbation mechanism. In the local attraction mechanism, each data point inside each cluster is attracted to the best element found in the cluster; this operation can be considered as an exploitation operator inside the cluster. The local perturbation modifies each data point to increase the exploitation process inside the cluster.

### 2.2.1. Local Attraction

For this operation, it is assumed that  $c_q^k$  represents a cluster and it is composed of a set of  $|c_q^k|$  data points  $\mathbf{d}_i^k (i \in c_q^k)$ , and that each element  $\mathbf{d}_i^k$  is attracted to the best data point  $\mathbf{d}_b^k \{d_{b,1}^k, d_{b,2}^k, \dots, d_{b,m}^k\}$  of the cluster based on the fitness value it represents. In this paper, the best element is considered as the minimum argument which minimizes the following objective function:

$$\mathbf{d}_b^k = \min_{i \in c_q^k} J(\mathbf{d}_i^k) \tag{3}$$

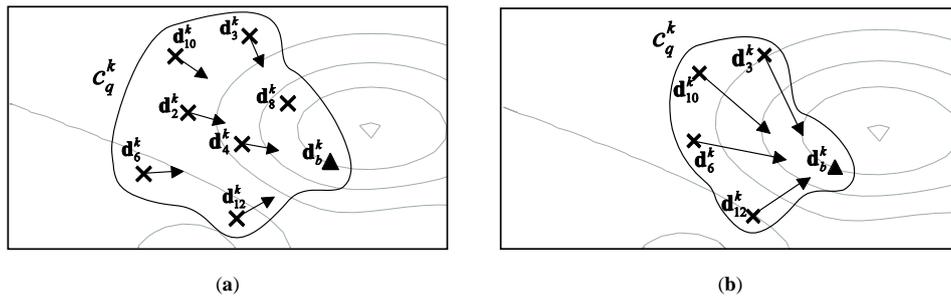
Then, the local attraction operator will modify the dimensionality of the  $\mathbf{d}_i^k$  data point as follows:

$$d_{i,j}^{k+1} = d_{i,j}^k + \rho_{c_q^k} \cdot z \cdot (d_{b,j}^k - d_{i,j}^k) \tag{4}$$

where  $z$  corresponds to a chaotic sequence value obtained by the Iterative Chaotic Map with Infinite Collapses (ICMIC) [37] chaotic map, to generate near-uniform distribution to maintain diversity among the population [38]. The density term  $\rho_{c_q^k}$  is then calculated as follows:

$$\rho_{c_q^k} = \frac{|c_q^k|}{N_D} \tag{5}$$

The cluster density term  $\rho_{c_q^k}$  is the quotient between the number of data points belonging to a given cluster  $|c_q^k|$  and the number of solutions in the population  $N_D$ . This quotient will produce lower values (low-density) when the number of elements belonging to a given cluster  $|c_q^k|$  contains few data points. In contrast, the density term  $\rho_{c_q^k}$  will produce higher values (high-density) when the number of elements belonging to a given cluster  $|c_q^k|$  contains a higher number of data points. The density term  $\rho_{c_q^k}$  then acts as an attraction factor among the data points of a given cluster and their corresponding best cluster elements  $\mathbf{d}_b^k$ . The induced effect of the density term  $\rho_{c_q^k}$  implies two different scenarios: (i) clusters containing a high number of data points, whereby the quotient  $\frac{|c_q^k|}{N_D}$  will produce high-density values which will obtain larger movements by Equation (4). The direct effect of this is large movements towards the best cluster element, improving the exploration of the inner space of the cluster, but smaller search capabilities in the exploitation of the inner space of the cluster. To illustrate this effect, Figure 1a represents the case when the cluster contains a high number of elements; hence, the quotient  $\frac{|c_q^k|}{N_D}$  will produce high-density values, and Equation (4) will produce large movements inside the cluster. The arrows in the Figure, represent smaller search capabilities in the exploitation of the inner space of the cluster. (ii) Clusters containing few datapoints, whereby the quotient  $\frac{|c_q^k|}{N_D}$  will produce low-density values which will obtain smaller movements by Equation (4). The direct effect of this is small movements towards the best cluster element, not improving the exploration of the inner space of the cluster, but producing larger search capabilities in the exploitation of the inner space of the cluster. To illustrate this effect, Figure 1b represents the case when the cluster contains few elements; hence, the quotient  $\frac{|c_q^k|}{N_D}$  will produce low-density values, and Equation (4) will produce small movements inside the cluster but larger search capabilities to exploit the inner space of the cluster. The arrows in the Figure represent larger search capabilities in the exploitation of the inner space of the cluster.



**Figure 1.** Local attraction operation in two scenarios: (a) cluster containing several elements (b) cluster containing few elements. The arrows represent (a) small search capabilities (low exploitation, high exploration) inside the cluster and (b) large search capabilities (high exploitation, low exploration) inside the cluster.

2.2.2. Local Perturbation

In this operation, the resulting repositioned solutions by the local attraction mechanism are perturbed inside the clusters to improve its exploitation search. Each produced solution by the attraction method is then modified to conduct the search strategy inside each cluster. According to this procedure, the produced element  $d_i^{k+1}$  generates two different subelements:  $h_i^A = \{h_{i,1}^A, \dots, h_{i,n}^A\}$  and  $h_i^B = \{h_{i,1}^B, \dots, h_{i,n}^B\}$ . Both elements are generated based on:

$$h_{i,j}^A = d_{i,j}^{k+1} + (d_{i,j}^{k+1} \cdot z_A \cdot v_A); h_{i,j}^B = d_{i,j}^{k+1} - (d_{i,j}^{k+1} \cdot z_B \cdot v_B) \tag{6}$$

where  $z_A$  and  $z_B$  represent chaotic values generated by an ICMIC chaotic map. The terms  $v_A$  and  $v_B$  correspond to a radial neighborhood described as follows:

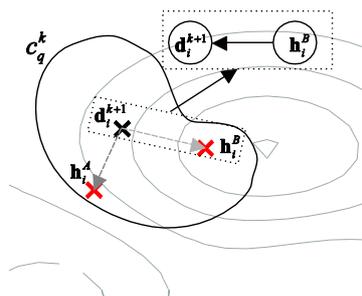
$$v_l = \cos(\alpha \cdot r); l = A, B \tag{7}$$

where  $r$  is a random number in the range  $[0, 2\pi]$ , and  $\alpha$  corresponds to the self-adaptive value described in [29].

The last step in this operation, is the elitist selection among  $d_i^{k+1}$ ,  $h_i^A$  and  $h_i^B$  to hold only the best elements from each generation. This process is described as follows:

$$d_i^{k+1} = \begin{cases} h_i^A & \text{if } J(h_i^A) < J(d_i^{k+1}) \\ h_i^B & \text{if } J(h_i^B) < J(d_i^{k+1}) \\ d_i^{k+1} & \text{otherwise} \end{cases} \tag{8}$$

To graphically illustrate the local perturbation operation, Figure 2 summarizes the procedure. The figure shows that element  $h_i^B$  presents a better fitness value than its predecessor  $d_i^{k+1}$ , and that under such a condition,  $h_i^B$  replaces  $d_i^{k+1}$  in a future iteration.



**Figure 2.** The effect of local perturbation operation.

### 2.3. Extracuster Operation

This CCO operation improves the global search. The operation considers two different parts: Global attraction and global perturbation. In global attraction, the best elements of each cluster are attracted to the best global solution which has occurred so far. In global perturbation, the repositioned elements produced by the global attraction movement are perturbed to increase the search capabilities of the method. This extracuster mechanism establishes a balance between the exploration and exploitation stages.

#### 2.3.1. Global Attraction

This operation moves the best elements of each cluster  $\mathbf{d}_b^k$  towards the best element to have occurred so far in the entire optimization process  $\mathbf{d}_B^k$ . This global operator is described as follows:

$$d_{b,j}^{k+1} = d_{b,j}^k + (d_{B,j}^k - d_{b,j}^k) \cdot rand(\cdot) \cdot v_G \tag{9}$$

where  $rand(\cdot)$  represents a random number between  $[0, 1]$  and  $v_G$  corresponds to a radial neighborhood from Equation (7).

#### 2.3.2. Global Perturbation

After the application of global attraction, the repositioned data points will produce two solutions in terms of radial movement. The aim of this procedure is to increase the exploitation rate of the search mechanism outside the clusters but inside the search space. In this operation, two different elements  $\mathbf{h}_b^R = \{h_{b,1}^R, \dots, h_{b,n}^R\}$  and  $\mathbf{h}_b^S = \{h_{b,1}^S, \dots, h_{b,n}^S\}$  are obtained as follows:

$$h_{b,j}^R = d_{b,j}^{k+1} + (d_{b,j}^{k+1} \cdot r_R \cdot v_R); h_{b,j}^S = d_{b,j}^{k+1} - (d_{b,j}^{k+1} \cdot r_S \cdot v_S) \tag{10}$$

where  $r_R$  and  $r_S$  correspond to random numbers between  $(0, 1)$ .  $v_R$  and  $v_S$  are radial neighborhoods generated by Equation (7).

The last step in this operation is an elitist selection among  $\mathbf{d}_i^{k+1}$ ,  $\mathbf{h}_b^R$  and  $\mathbf{h}_b^S$  to keep only the best elements from each generation. This process is described as follows:

$$\mathbf{d}_b^{k+1} = \begin{cases} \mathbf{h}_b^R & \text{if } J(\mathbf{h}_b^R) < J(\mathbf{d}_b^{k+1}) \\ \mathbf{h}_b^S & \text{if } J(\mathbf{h}_b^S) < J(\mathbf{d}_b^{k+1}) \\ \mathbf{d}_b^{k+1} & \text{otherwise} \end{cases} \tag{11}$$

To illustrate the extracuster operation, Figure 3 summarizes both the global attraction mechanism and the global perturbation procedure.

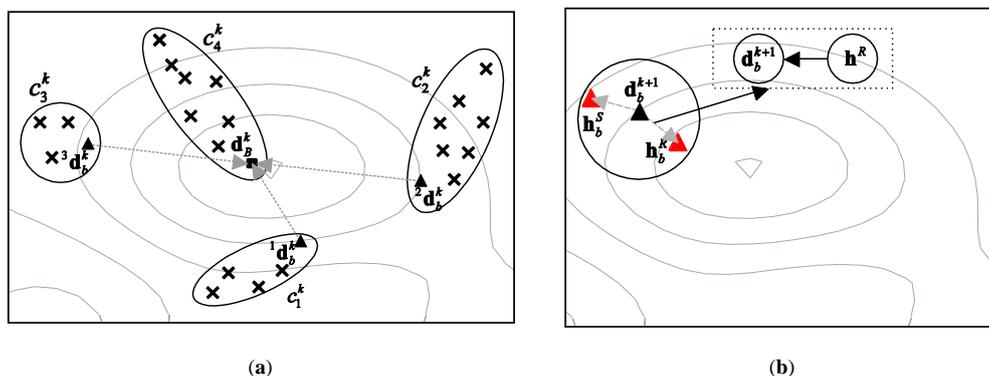


Figure 3. Extracuster procedure for (a) global attraction and for (b) global perturbation.

The following pseudo-code in Algorithm 1 summarizes the entire iteratively process of the CCO.

**Algorithm 1.** Pseudo-code for the Cluster-Chaotic-Optimization (CCO) algorithm.

---

```

1.   Input:  $N_D, gen, k = 0$ 
2.    $\mathbf{D}^k \leftarrow \text{InitializePopulation}(N_D);$ 
3.   while  $k \leq gen$  do
4.      $\mathbf{d}_B^k \leftarrow \text{SelectBestParticle}(\mathbf{D}^k);$ 
5.      $[c_q^k, g] \leftarrow \text{WardClustering}(\mathbf{D}^k);$ 
6.      $\alpha \leftarrow \text{CalculatePerturbation}(gen);$  Self-Adaptive value from [29]
7.     for ( $q = 1; q \leq g; q++$ )
8.        $\mathbf{d}_b^k \leftarrow \text{SelectBestCluster}(c_q^k);$ 
9.        $\mathbf{d}_l^{k+1} \leftarrow \text{LocalAttractionOperation}(c_q^k); l \in c_q^k$  Intracluster
10.       $\mathbf{d}_l^{k+1} \leftarrow \text{LocalPerturbationOperation}(\mathbf{d}_l^{k+1});$  procedure
11.     end for
12.     for (each element of  $c_q^k$ )
13.        $\mathbf{d}_b^{k+1} \leftarrow \text{GlobalAttractionOperation}(\mathbf{d}_b^k);$  Extracluster
14.        $\mathbf{d}_b^{k+1} \leftarrow \text{GlobalPerturbationOperation}(\mathbf{d}_b^{k+1});$  procedure
15.     end for
16.      $k = k + 1;$ 
17.   end while
18.   Output:  $\mathbf{d}_B^k$ 

```

---

### 3. Multimodal Cluster-Chaotic-Optimization (MCCO)

In CCO, the optimization process is driven by the application of a data analysis technique with chaotic perturbations. CCO divides the population considering the spatial information among individuals. This clustering process is based on the computation of a hierarchical tree, where the natural associations among each data point (individual) are determined. The clustering method called “Ward” is cataloged as a hierarchical clustering methodology, where each element starts by forming a single cluster, and then an association tree is generated over the remaining elements. This tree structure is used to define the level at which the clustering algorithm will produce clusters. The formed clusters share similarities among data points. CCO uses the idea of clustering from the Ward method to partition the population into similar groups at each iteration of the optimization process. In the beginning, each element is treated as a single cluster; then, during the application of its evolutionary operators, it starts grouping clusters containing more elements. Then, CCO operates each cluster differently by exploring and exploiting inside and outside each cluster.

The CCO uses the intracluster operation to locally explore and exploit inside each formed cluster. This process is achieved by two operations: local attraction and local perturbation. Under local attraction, each element of a given cluster is attracted to the best element in such a cluster. The way each element moves towards the best individual in the cluster is based on the density measure of the cluster. In the CCO, the density of a cluster refers to the number of elements a cluster has. Clusters containing few elements will attract each element quickly. On the other hand, if a cluster contains a high number of elements, the elements will attract each other slowly. This process can be defined as an exploration operator inside the cluster. To maintain a balance between exploration and exploitation inside the clusters, CCO defines a local perturbation operator as an exploitation mechanism inside the clusters. In this way, two different solutions are radially generated to improve the search mechanism.

On the other hand, CCO uses the extracuster operation to globally explore and exploit outside the clusters, that is, in the overall search space. To accomplish this, CCO considers both the global attraction and global perturbation operations. Under global attraction, the best elements of each cluster are attracted to the global best solution to have occurred so far. This procedure improves the exploitation stage outside each cluster but inside the feasible search space of a given optimization

problem. Then, a redefinition phase is computed. The CCO uses global perturbation as an exploitation operator outside the clusters but inside the search space. This whole process maintains the population diversity and promotes a balance among the evolutionary exploitation and exploration stages.

The spatial associations from the CCO operators suggest an inherent multimodal behavior. Each time an iteration begins, a clustering procedure is executed. This mechanism will group each individual in the population into similar individuals forming clusters. This clustering process suggests the agglomeration of individuals into potential search zones. Under such circumstances, CCO presents a certain degree of multimodality in its operators. However, the original CCO structure is not suitable for detecting multiple optima in a single run. Under such limitations, CCO can be adapted to incorporate multimodal capabilities.

In this paper, a multimodal extension of the CCO is presented to incorporate multimodal capabilities into the original structure of the CCO. The concept of dominance commonly found in social animal associations presents the nature-inspired structure called “competitive memory”. Under the competitive memory approach, each individual will confront its neighbors. The resulting individual will be catalogued as a potential optimum. Then, an updating scheme will manage the diversity of the population. The effect of this computational structure will provide a multimodal structure where optimal and suboptimal solutions will be carried out at each iteration.

The multimodal extension used to incorporate multimodal capabilities in the CCO was conceived based on the concept of dominance in animal interactions. Biologists have demonstrated that social interactions among animals remain in an animal’s memory. Such a structure has been called “competitive memory” [31–33]. In this structure, it is established that previous group interactions can affect social interactions in the future in terms of aggressiveness. Such aggressiveness keeps two or more individuals as distant as possible from one another, where the most dominant individual prevails as the other withdraws. From a computational point of view, the idea of dominance among individuals in a population is implemented based on a data structure called competitive memory.

To implement the competitive memory approach, two types of memory must be generated: historic  $\mathbf{M}_H$  and population memories  $\mathbf{M}_P$ . Historic memory stores promissory solutions through the optimization process, in contrast to population memory, which only stores the solutions for each generation. Once these memory structures have been initialized, competition and update mechanisms are required.

### 3.1. Initialization Phase

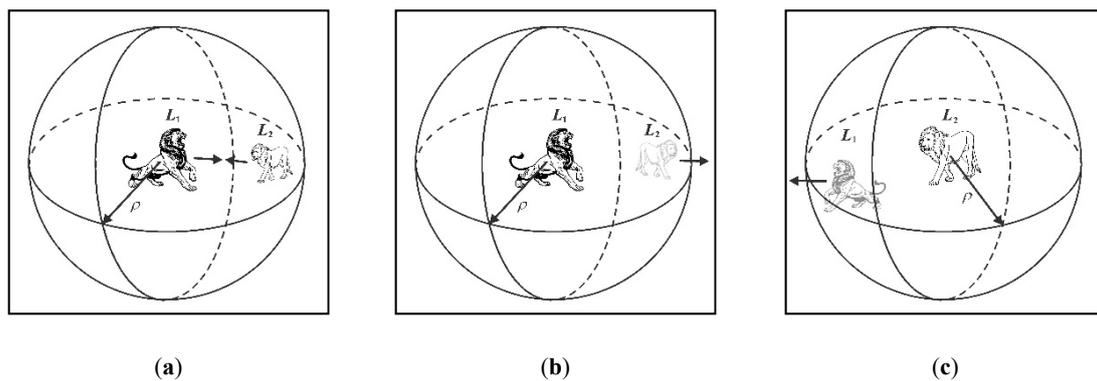
The first step considered for the implementation of the competitive memory approach in the CCO is the initialization of the memory mechanism. For that, once the initialization procedure from Section 2.1 has been executed, a sorted copy of the population will create the historic memory  $\mathbf{M}_H = \{\mathbf{m}_H^1, \mathbf{m}_H^2, \dots, \mathbf{m}_H^n\}$ , where each  $\mathbf{m}_H^i$  vector corresponds to an element belonging to the historic memory. A sorted copy of the population will also create the population memory  $\mathbf{M}_P = \{\mathbf{m}_P^1, \mathbf{m}_P^2, \dots, \mathbf{m}_P^n\}$ , where each element  $\mathbf{m}_P^i$  corresponds to an individual stored in the population memory. After the initialization process has occurred, the population memory will be affected by the CCO evolutionary operators and the historic memory will maintain potential optima during the optimization process.

### 3.2. Competition Phase

This procedure is based on the biological concept of dominance. Animal dominance is a social interaction behavior among two animals. Animals maintain a distance from each other to avoid confrontation. The distance is based on how aggressively the animals behave. When two animals confront each other inside a radius distance, the most dominant animal will prevail, while the other withdraws. In order to implement this idea, a set of competition rules must be applied for each solution to be part of the  $\mathbf{M}_H$ . The competition rules are based on the distance ( $\delta$ ) and fitness values among  $\mathbf{M}_P^k$  and  $\mathbf{M}_H$ . The following rules are considered in the implementation of the competition phase in MCCO.

1. Compute the dominance radius  $\rho$ .
2. Compute the euclidean distance  $\delta$  among elements of  $\mathbf{M}_H$  and elements of  $\mathbf{M}_P^k$ .
3. If the distance  $\delta$  between two individuals is less than the dominance radius  $\rho$ , then the prevailing individuals belonging to  $\mathbf{M}_H$  will be stored in a temporary historic memory  $\mathbf{T}_H$ , while the prevailing individuals in  $\mathbf{M}_P^k$  will be stored in temporary population memory  $\mathbf{T}_P$ .
4. The temporary memory structure  $\mathbf{T}$  will be the union of  $\mathbf{T}_H$  and  $\mathbf{T}_P$ .

To illustrate the previously described competition rules, Figure 4 graphically illustrates the competition phase between  $L_1$  and  $L_2$  animals using the representation  $L_1$  as individual  $\mathbf{m}_i^H$  stored in the  $\mathbf{M}_H$ , and  $L_2$  to represent individual  $\mathbf{m}_j^P$  stored in  $\mathbf{M}_P^k$ .



**Figure 4.** Competition phase (a) when  $L_2$  enters to the radius of  $L_1$ , (b)  $L_1$  remains unbeaten (c)  $L_2$  remains unbeaten.

In Figure 4a,  $L_2$  enters to the radius of  $L_1$ ; then, they confront each other, considering their fitness values. If  $L_2$  possesses a lower fitness value than  $L_1$ , then  $L_1$  remains unbeaten and  $L_2$  is removed from  $\mathbf{M}_P^k$  (Figure 4b). On the other hand, if  $L_1$  possesses a lower fitness value than  $L_2$ ,  $L_2$  remains unbeaten and  $L_1$  is removed from  $\mathbf{M}_H$  (Figure 4c).

The dominance radius  $\rho$  in Figure 4 is computed by:

$$\rho = \frac{\prod_{j=1}^d (u_j - l_j)}{\kappa \cdot d} \tag{12}$$

where  $l_j$  and  $u_j$  correspond the lower and upper limits for the  $j$ -th decision variable, respectively.  $\kappa$  corresponds to a proportional factor to adjust the radius to a minimum value regarding the number of dimensions for the objective function. The  $\kappa$  parameter was experimentally configured to 20. Such an experimental value was chosen considering the sensitivity analysis reported in Section 4.3. The pseudo-code for the competition phase is summarized in Algorithm 2.

### 3.3. Update Phase

Finally, a mechanism to maintain population diversity in the optimization process is considered in the last step. The updating scheme for the competitive memory approach aims to obtain the historic memory  $\mathbf{M}_H$  for future iterations. The historic memory will contain and maintain the best solutions through the optimization process. In the competition phase, a temporary historic memory  $\mathbf{T}_H$  is created; however, the number of its elements could be smaller than the historic memory  $\mathbf{M}_H$ . Hence, if the aforementioned condition is satisfied ( $|\mathbf{T}_H| < |\mathbf{M}_H|$ ), then the update phase is executed considering two scenarios as:

1. If  $|\mathbf{T}_P| > 0$ , then the best individuals belonging to  $\mathbf{T}_P$  will be stored in  $\mathbf{T}_H$ .

- If  $|\mathbf{T}_H| < N_D$ , then the best solutions in  $\mathbf{P}^k$  will be allocated to  $\mathbf{T}_H$ .

---

**Algorithm 2.** Pseudo-code for the competition phase.

---

```

1.  $\mathbf{T}_H \leftarrow \emptyset$ 
2.  $\mathbf{T}_P \leftarrow \emptyset$ 
3.  $\rho \leftarrow \frac{\prod_{j=1}^d (u_j - l_j)}{\kappa \cdot d}$  1st Rule
4. for ( $i = 1; i \leq \text{size}(\mathbf{M}_H); i++$ )
5.   for ( $j = 1; j \leq \text{size}(\mathbf{M}_P); j++$ )
6.      $\delta \leftarrow \sqrt{\sum_{z=1}^d (\mathbf{m}_{i,z}^H - \mathbf{m}_{j,z}^P)^2}$  2st Rule
7.     if  $\delta < \rho$  3rd Rule
8.       if  $J(\mathbf{m}_{i,z}^H) < J(\mathbf{m}_{j,z}^P)$ 
9.          $\mathbf{T}_H \leftarrow \mathbf{T}_H \cup \mathbf{m}_{i,z}^H$ 
10.      else 4rd Rule
11.         $\mathbf{T}_P \leftarrow \mathbf{T}_P \cup \mathbf{m}_{j,z}^P$ 
12.      end if
13.    end if
14.  end for
15. end for
16.  $\mathbf{T} = \mathbf{T}_H \cup \mathbf{T}_P$ 

```

---

To summarize the update scheme, Algorithm 3 presents pseudo-code for the previous description.

---

**Algorithm 3.** Pseudo-code for the update phase.

---

```

1. if  $\text{size}(\mathbf{T}_H) < \text{size}(\mathbf{M}_H)$ 
2.   if  $\text{size}(\mathbf{T}_P) > 0$ 
3.      $\mathbf{T}_H \leftarrow \mathbf{T}_H \cup \mathbf{T}_P$ 
4.   end if
5.   if  $\text{size}(\mathbf{T}_H) < N_D$ 
6.      $\mathbf{T}_H \leftarrow \mathbf{T}_H \cup \mathbf{P}^k$ 
7.   end if
8. end if
9.  $\mathbf{M}_H \leftarrow \mathbf{T}_H$ 

```

---

### 3.4. The Complete Multimodal Cluster-Chaotic-Optimization (MCCO)

To incorporate multimodal capabilities into the original structure of CCO, the MCCO requires three operators to allocate and manage the potential optima: initialization, competition, and update. In the initialization phase, a memory mechanism is initialized based on the current population; in this process, two types of memories are generated in order to computationally abstract the concept of dominance. Then, in the competition phase, the solutions in both memories confront each other in order to determine the most dominant solutions. Finally, the updating scheme manages the historic memory to produce a new population to be used in future iterations. Under the complete memory mechanism of the MCCO, potential optima will be stored and maintained during the whole optimization process by holding only the solutions which present better fitness values. The complete pseudo-code for the MCCO algorithm can be summarized in Algorithm 4.

In Algorithm 4, the original operators of the CCO are extended with the memory mechanism described in Section 3; the memory initialization process is achieved by line 3. Then, the competitive

phase is applied in line 17, and the update phase is accomplished in line 18. The original structure of the intra- and extra- cluster operations are found in lines 8–12 and 13–16, respectively.

---

**Algorithm 4.** Pseudo-code for the Multimodal Cluster-Chaotic-Optimization (MCCO) algorithm.

---

```

1.   Input:  $N_D, gen, k=0$ 
2.    $D^k \leftarrow \text{InitializePopulation}(N_D);$ 
3.    $[M_H, M_P] \leftarrow \text{InitializeMemory}(D^k);$                                 Memory initialization
4.   while  $k \leq gen$  do
5.      $d_B^k \leftarrow \text{SelectBestParticle}(D^k);$ 
6.      $[c_q^k, g] \leftarrow \text{WardClustering}(D^k);$ 
7.      $\alpha \leftarrow \text{CalculatePerturbation}(gen);$ 
8.     for ( $q = 1; q \leq g; q++$ )
9.        $d_b^k \leftarrow \text{SelectBestCluster}(c_q^k);$ 
10.       $d_l^{k+1} \leftarrow \text{LocalAttractionOperation}(c_q^k); l \in c_q^k$ 
11.       $d_l^{k+1} \leftarrow \text{LocalPerturbationOperation}(d_l^{k+1});$ 
12.    end for                                Original CCO operators
13.    for (each element of  $c_q^k$ )
14.       $d_b^{k+1} \leftarrow \text{GlobalAttractionOperation}(d_b^k);$ 
15.       $d_b^{k+1} \leftarrow \text{GlobalPerturbationOperation}(d_b^{k+1});$ 
16.    end for
17.     $[T_H, T_P] \leftarrow \text{CompetitionPhase}(M_H, M_P);$                                 Memory Competition and Update
18.     $[M_H, M_P] \leftarrow \text{UpdatePhase}(M_H, M_P, T_H, T_P);$ 
19.     $k = k + 1;$ 
20.     $D^k \leftarrow M_P$ 
21.  end while
22.  Output:  $d_B^k, D^k$ 

```

---

## 4. Experimental Results

This section presents a numerical comparison among MCCO and 11 state-of-the-art multimodal techniques. The performance results were obtained by the evaluation of 23 multimodal benchmark functions containing different types of complexities. The experimental analysis was based on the computation of commonly used performance metrics in the multimodal literature. Such metrics measure the ability of each multimodal methodology to quantify the number of approximated solutions considering true optima. In the Section 4.1, each of the performance metrics used in the experimental study is described. Section 4.2 presents the analytical methodology considered in this study to obtain the true optimal values for each multimodal benchmark function. Section 4.3 presents the numerical results of MCCO, and the rest of the multimodal approaches are compared considering the performance metrics.

### 4.1. Performance Metrics

In this section, six multimodal optimization performance indexes are presented. The set of metrics is composed of the Effective Peak Number (EPN), the Maximum Peak Ratio (MPR), the Peak Accuracy (PA), the Distance Accuracy (DA), the Peak Ratio (PR), and the Success Rate (SR). The entire set of metrics was used extensively to quantify the performance of many multimodal optimization techniques [17,39–41]. The entire set of metrics expresses the performance of a multimodal approach based on the difference among true optima and the approximated optimal values. The EPN reflects the capability of a multimodal technique to obtain most of the optima. The MPR computes the consistency of the approximated optima over true optima. On the other hand, PA measures the error among approximated optima and true optima. Similarly, DA indicates the total error, considering each independent variable of the objective function. PR calculates the percentage of the total number of approximated optima over multiple executions of a given algorithm. Lastly, SR measures the

successful percentage of runs considering the total number of executions. The following paragraphs mathematically describe each metric.

*Effective Peak Number (EPN)*. This metric quantifies the number of approximated solutions identified as valid optima. Each approximated solution  $\hat{\mathbf{o}}$  is considered as a valid optimum if the Euclidean distance between  $\hat{\mathbf{o}}$  and the true optimum  $\mathbf{o}$  is less than  $\mu$ . The EPN is calculated as follows:

$$EPN = \|\mathbf{o}_i - \hat{\mathbf{o}}_j\| < \mu \tag{13}$$

where the subindexes  $i$  and  $j$  correspond to the  $i$ -th and the  $j$ -th true optimum and the approximated optimum, respectively. Additionally,  $\mu$  is a threshold value which refers to the accuracy. The value of  $\mu$  was set to 0.5. This value corresponds to the accuracy level in [41].

*Maximum Peak Ratio (MPR)*. This metric computes the consistency of the approximated optima over true optima. MPR is defined as:

$$MPR = \frac{\sum_{j=1}^{EPN} J(\hat{\mathbf{o}}_j)}{\sum_{i=1}^O J(\mathbf{o}_i)} \tag{14}$$

where  $EPN$  and  $O$  correspond to the number of valid optima and the number of true optima, respectively.

*Peak Accuracy (PA)*. Calculates the obtained error among approximated optima and true optima as follows:

$$PA = \sum_{i=1}^O \|J(\mathbf{o}_i) - J(\hat{\mathbf{o}}_i)\| \tag{15}$$

*Distance Accuracy (DA)*. Since the calculated error in PA is based on the fitness value, it does not consider the closeness over peaks. For that, DA computes the error between approximated optima and true optima according to the following model:

$$DA = \sum_{i=1}^O \|\mathbf{o}_i - \hat{\mathbf{o}}_i\| \tag{16}$$

*Peak Ratio (PR)*. PR calculates the percentage of the total number of approximated optima over multiple executions of a given algorithm as follows:

$$PR = \frac{\sum_{i=1}^{NR} EPN_i}{O \cdot NR} \tag{17}$$

*Successful Rate (SR)*. Measures the successful percentage of runs considering the total number of executions as:

$$SR = \frac{NSR}{NR} \tag{18}$$

where  $NSR$  corresponds to the number of successful runs and  $NR$  denotes the total number of executions.

#### 4.2. True Optima Determination

Each of the previously described multimodal metrics operates considering the true optima solutions for each benchmark function. Under such circumstances, true optimal values are required. Most of the reported literature on multimodal optimization lacks information related to the numerical values of true optima. In this paper, the calculation process to obtain the numerical values of each optimum is based on derivate application. To obtain all the optima values, the middle point between the highest and lowest values is defined for each benchmark function. Then, all the optima found below (in case of minimization) the middle point will be target optima. Under the target optima,

the application of the second partial derivative is required to analytically compute the optimal values. The following model describes the true optimal set  $\mathbf{T}$ .

$$\mathbf{T} = \{\mathbf{o} \in J \mid \mathbf{o} \leq m\} \quad (19)$$

where  $J$  corresponds to the objective function,  $\mathbf{o}$  is the optimum, and  $m$  represents the middle point used as a threshold value to compute optimal values. Then, Equation (20) defines the second partial derivative discriminant:

$$D \equiv J_{xx}J_{yy} - J_{xy}J_{yx} \quad (20)$$

To indicate if certain point  $(x_0, y_0)$  could represent a local minimum, the discriminant  $D$  is used as follows:

$$\mathbf{o} = \{D > 0, J_{xx}(x_0, y_0) > 0\} \quad (21)$$

From Equation (21), it can be shown that the process to obtain local minima is based on a minimization process.

#### 4.3. Performance Comparison

In this section, the numerical results of the proposed MCCO are presented by comparing the performance among MCCO and 11 state-of-the-art multimodal approaches, considering a set of 14 multimodal benchmark functions. The benchmark functions have been widely used to test the multimodal capabilities of several multimodal functions [41–43]. Tables A1 and A2 in Appendix A mathematically describe the test functions considered in the experimental study. For clarity, the benchmark functions have been split into two tables. Table A1 describes functions  $J_1$ – $J_7$ , and Table A2 describes functions  $J_8$ – $J_{14}$ . In the tables, the features of each benchmark function are defined. The search domain column indicates the box constraints for each objective function,  $n$  corresponds to the dimensionality tested, and the optima number corresponds to the number of true optima determined by the second partial derivative method from Section 4.2.

For comparison purposes, the MCCO is compared against 11 multimodal methodologies: Locally Informed Particle Swarm Model (LIPSM) [44], Fitness Sharing Differential Evolution (FSDE) [17], Clonal Selection Algorithm (CSA) [45], Deterministic Crowding Genetic Algorithm (DCGA) [46,47], Locally Informative Niching Differential Evolution (LoINDE) [15], Proximity-based Crowding DE (PNPCDE) [48], Multimodal Gravitational Search algorithm (MGSA) [49], History-based Topological Speciation (HTS) [22], Multiobjective Optimization for Multiple optimal of Multimodal Optimization algorithm (MOMMOP) [28], Ensemble and Arithmetic Recombination-Based Speciation DE (EARSDE) [50], and Region-based Memetic algorithm (RM) [51].

The comparison scheme involves the evaluation of the six multimodal metrics described in Section 4.1. Also, it is considered a statistical validation framework based on a rank sum [52] test to avoid the random effect. The population size has been configured to 100 individuals, and the maximum number of iterations has been configured as 500, considering 30 independent runs. Each optimization process is executed using MATLAB® R2018b, Windows-7 OS, x64-based PC, Intel(R) Core-i7(R)-CPU, 2.20 GHz with 16 GB RAM. The initial configuration parameters for each multimodal approach were devised according to the guidelines in Table 1. These configuration settings were chosen since they represent the best parameters for each multimodal approach according to their reported guidelines.

Additionally, the  $\kappa$  parameter was experimentally configured to 20; this value was chosen by the sensitivity analysis shown in Table 2. In the table, an evaluation of the MCCO method is reported for each benchmark function considering the EPN metric. The sensitivity analysis was conducted on 30 independent runs. The best entries in the table are in bold, and the numbers in parenthesis are the standard deviations.

Tables 3 and 4 present the numerical results from the experimental study of all multimodal approaches. To make a clear representation of the numerical results, Table 3 reports the experimental

results for functions  $J_1$ – $J_7$ , and Table 4 reports the numerical results for functions  $J_8$ – $J_{14}$ . In the tables, the numerical values for each metric are presented. Additionally, to measure the computational effort of each multimodal approach, the Number of Function Calls (NFC) and the execution time (T) in seconds were assessed. Finally, the entries in parentheses are the standard deviations of each particular metric.

**Table 1.** Parameter configuration for each multimodal method used in the experimental study.

Algorithm	Parameter(s)	Reference
DCGA	Crossover probability $cp = 0.9$ , Mutation probability $mp = 0.1$	[46]
CSA	Mutation probability $mp = 0.01$ , percentile to random reshuffle $per = 0.0$ , clone per candidate $fat = 0.1$	[45]
FSDE	Crossover probability $cr = 0.9$ , differential weight $dw = 0.1$ , sharing radius $\sigma_{share} = 0.1$ , $\alpha = 1.0$	[46]
LIPSM	Neighborhood size $nsize = 2$	[44]
LoINDE	Crossover probability $cr = 0.2$ , differential weight $dw = 0.9$	[15]
MGSA	Final percentage $fp = 0.02$	[49]
PNPCDE	Crossover probability $cr = 0.2$ , differential weight $dw = 0.9$	[48]
HTS	Crossover probability $cr = 0.9$ , differential weight $dw = 0.1$	[22]
MOMMOP	Crossover probability $cr = 0.7$ , differential weight $dw = 0.5$	[28]
EARSDE	Crossover probability $cr = 0.9$ , differential weight $dw = 0.1$ , sharing radius $\sigma_{share} = 0.1$ , $\alpha = 1.0$	[50]
RM	Detail parameters are described by guidelines of the author.	[51]

**Table 2.** Sensitivity analysis of the  $\kappa$  parameter.

Function	$\kappa = 10$	$\kappa = 15$	$\kappa = 20$	$\kappa = 25$	$\kappa = 30$
$J_1$	3.0000 (0.7071)	3.4000 (0.8944)	<b>6.0000</b> <b>(0.0000)</b>	5.4000 (0.8944)	4.6000 (0.5477)
$J_2$	1.0000 (0.7071)	2.2000 (0.8367)	<b>4.0000</b> <b>(0.0000)</b>	3.8000 (0.4472)	3.6000 (0.5477)
$J_3$	2.4000 (0.5477)	4.4000 (1.8166)	<b>12.0000</b> <b>(0.0000)</b>	6.8000 (1.3038)	6.4000 (1.9494)
$J_4$	10.8000 (1.6432)	16.2000 (2.0494)	<b>19.0000</b> <b>(2.4083)</b>	18.8000 (0.4472)	13.2000 (1.0954)
$J_5$	1.2000 (0.4472)	3.4000 (0.8944)	<b>5.0000</b> <b>(0.0000)</b>	4.4000 (0.5477)	4.0000 (1.2247)
$J_6$	2.8000 (0.4472)	3.0000 (0.0000)	<b>3.0000</b> <b>(0.0000)</b>	3.0000 (0.0000)	3.0000 (0.0000)
$J_7$	4.0000 (0.0000)	4.0000 (0.0000)	<b>4.0000</b> <b>(0.0000)</b>	3.6000 (0.5477)	3.0000 (0.7071)
$J_8$	8.0000 (0.0000)	7.8000 (0.4472)	<b>8.0000</b> <b>(0.0000)</b>	8.0000 (0.0000)	6.2000 (1.0954)
$J_9$	6.0000 (0.0000)	6.0000 (0.0000)	<b>6.0000</b> <b>(0.0000)</b>	6.0000 (0.0000)	6.0000 (0.0000)
$J_{10}$	15.2000 (2.8636)	20.0000 (1.5811)	<b>24.0000</b> <b>(0.7071)</b>	21.2000 (2.3875)	22.0000 (1.0000)
$J_{11}$	14.8000 (1.7889)	20.8000 (2.2804)	<b>24.0000</b> <b>(0.0000)</b>	22.2000 (1.3038)	20.4000 (1.1402)
$J_{12}$	3.6000 (0.8944)	6.4000 (0.8944)	<b>8.0000</b> <b>(0.0000)</b>	6.4000 (0.8944)	6.8000 (1.0954)
$J_{13}$	4.0000 (0.0000)	4.0000 (0.0000)	<b>4.8000</b> <b>(0.4472)</b>	4.0000 (0.0000)	4.0000 (0.0000)
$J_{14}$	4.2000 (1.3038)	6.4000 (0.5477)	<b>9.0000</b> <b>(0.0000)</b>	8.6000 (0.8944)	8.0000 (0.7071)

**Table 3.** Numerical results for  $J_1$ – $J_7$  multimodal test functions.

Function	Algorithm	EPN	MPR	PA	DA	PR	SR	NFC	T(s)
$J_1$	DCGA	3.7600 (1.2048)	0.6442 (0.1990)	126.2960 (69.4572)	11.9316 (5.7567)	6.47E−01	6.00E−02	5.2701e+06 (373.5616)	4.7476 (0.2630)
	CSA	1.8200 (0.3881)	−0.0107 (0.0017)	352.9702 (0.6908)	25.3819 (0.3957)	1.67E−01	2.34E−02	1.0020e+03 (0.0000)	3.6196 (0.4550)
	FSDE	3.0000 (0.0000)	0.8600 (0.0072)	54.4195 (2.4567)	14.7039 (0.0815)	4.93E−01	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	11.1163 (0.3280)
	LIPSM	3.9400 (1.2683)	0.6289 (0.2270)	130.8030 (78.5388)	12.7353 (6.2776)	6.53E−01	2.00E−02	<b>5.0000e+04</b> <b>(0.0000)</b>	<b>0.0715</b> <b>(0.0025)</b>
	LoINDE	0.1400 (0.3505)	−0.0327 (0.0909)	362.7803 (31.1560)	26.5336 (1.4380)	0.00E+00	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	6.3351 (0.1164)
	MGSA	1.0000 (0.0000)	−0.0174 (0.0819)	357.6663 (28.3122)	27.2754 (1.6772)	1.67E−01	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	14.8227 (0.0646)
	PNPCDE	3.0400 (0.1979)	−0.5547 (0.0376)	537.5972 (12.9847)	20.1959 (0.6167)	5.00E−01	4.31E−02	<b>5.0000e+04</b> <b>(0.0000)</b>	13.6878 (0.1993)
	HTS	3.9200 (0.8999)	1.1230 (0.2705)	249.6098 (59.7460)	14.6753 (2.5628)	6.90E−01	2.00E−02	4.7710e+06 (921,384.2807)	151.2031 (20.3608)
	MOMMOP	<b>6.0000</b> <b>(0.0000)</b>	0.7144 (0.1391)	153.3749 (40.9594)	6.0172 (1.2304)	<b>1.00E+00</b>	<b>1.00E+00</b>	<b>5.0000e+04</b> <b>(0.0000)</b>	19.9509 (0.2957)
	EARSDE	0.9800 (0.1414)	0.3004 (0.0448)	248.1227 (15.4322)	22.9300 (1.1826)	1.67E−01	0.00E+00	2.3418e+05 (103,335.0363)	7.4633 (1.2582)
	RM	5.2800 (0.6402)	0.7036 (0.1274)	107.5283 (43.2990)	8.3606 (3.8876)	8.90E−01	2.23E−01	<b>5.0000e+04</b> <b>(0.0000)</b>	8.2135 (0.7462)
	MCCO	<b>6.0000</b> <b>(0.0000)</b>	<b>0.9904</b> <b>(0.0075)</b>	<b>4.0639</b> <b>(2.6462)</b>	<b>0.6218</b> <b>(0.0963)</b>	<b>1.00E+00</b>	<b>1.00E+00</b>	<b>5.0000e+04</b> <b>(0.0000)</b>	4.5698 (0.7841)
$J_2$	DCGA	1.1200 (0.8485)	0.2685 (0.2015)	31.7400 (8.7399)	10.3789 (2.7592)	2.90E−01	2.00E−02	5.2700e+06 (394.2309)	4.5124 (0.1676)
	CSA	0.0000 (0.0000)	0.0000 (0.0000)	43.3904 (0.0000)	14.0472 (0.0000)	0.00E+00	0.00E+00	1.0020e+03 (0.0000)	4.1331 (0.3774)
	FSDE	<b>4.0000</b> <b>(0.0000)</b>	0.9887 (0.0077)	0.4906 (0.3318)	0.5365 (0.1768)	<b>1.00E+00</b>	<b>1.00E+00</b>	<b>5.0000e+04</b> <b>(0.0000)</b>	11.3112 (0.3865)
	LIPSM	1.7200 (1.2623)	0.4101 (0.2996)	25.5940 (12.9992)	8.4951 (4.0528)	5.00E−01	1.60E−01	<b>5.0000e+04</b> <b>(0.0000)</b>	<b>0.0712</b> <b>(0.0009)</b>
	LoINDE	2.8800 (0.9823)	0.0000 (0.0000)	43.3904 (0.0000)	10.3229 (1.2543)	7.00E−01	2.40E−01	<b>5.0000e+04</b> <b>(0.0000)</b>	6.4320 (0.0187)
	MGSA	1.0000 (0.0000)	0.1199 (0.0740)	38.1891 (3.2111)	15.0909 (2.1688)	2.50E−01	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	14.6761 (0.0265)
	PNPCDE	3.7000 (0.4629)	0.0000 (0.0000)	43.3904 (0.0000)	9.2659 (0.6119)	9.05E−01	6.20E−01	<b>5.0000e+04</b> <b>(0.0000)</b>	13.6181 (0.0343)
	HTS	1.5600 (0.9293)	0.3902 (0.2317)	26.4993 (10.0648)	13.3447 (2.4274)	3.50E−01	1.56E−02	4.0284e+06 (923,964.7574)	84.3728 (34.0255)
	MOMMOP	3.8200 (0.3881)	0.7118 (0.1288)	12.5069 (5.5905)	7.0579 (2.0677)	9.40E−01	7.60E−01	<b>5.0000e+04</b> <b>(0.0000)</b>	19.1834 (0.4985)
	EARSDE	1.0000 (0.0000)	0.2504 (0.0003)	32.5607 (0.0101)	13.6066 (1.5561)	2.50E−01	0.00E+00	2.0677e+05 (100,142.9924)	7.0750 (1.6465)
	RM	3.7400 (0.4431)	0.7180 (0.1452)	12.2383 (6.3022)	3.3264 (1.5125)	9.29E−01	5.00E−01	<b>5.0000e+04</b> <b>(0.0000)</b>	11.1452 (0.2781)
	MCCO	<b>4.0000</b> <b>(0.0000)</b>	<b>0.9891</b> <b>(0.0068)</b>	<b>0.4759</b> <b>(0.2947)</b>	<b>0.5273</b> <b>(0.1261)</b>	<b>1.00E+00</b>	<b>1.00E+00</b>	<b>5.0000e+04</b> <b>(0.0000)</b>	4.5961 (0.7636)
$J_3$	DCGA	1.4400 (1.1095)	0.1198 (0.0927)	9.9554 (1.0486)	130.1057 (13.7530)	1.20E−01	5.47E−02	5.2700e+06 (325.3758)	4.7424 (0.0041)
	CSA	0.0000 (0.0000)	0.0000 (0.0000)	11.3110 (0.0000)	147.5853 (0.0000)	0.00E+00	0.00E+00	1.0020e+03 (0.0000)	4.2001 (0.2209)
	FSDE	8.5800 (1.0708)	0.7167 (0.0880)	3.2041 (0.9956)	41.7794 (12.1795)	7.27E−01	4.75E−01	<b>5.0000e+04</b> <b>(0.0000)</b>	11.1469 (0.1650)

Table 3. Cont.

Function	Algorithm	EPN	MPR	PA	DA	PR	SR	NFC	T(s)
$J_3$	LIPSM	2.5200 (1.5418)	0.2094 (0.1279)	8.9427 (1.4471)	117.2757 (18.4672)	2.40E-01	3.12E-02	<b>5.0000e+04</b> <b>(0.0000)</b>	<b>0.0701</b> <b>(0.0017)</b>
	LoINDE	11.0800 (0.8533)	0.0000 (0.0000)	11.3110 (0.0000)	35.0425 (7.9412)	9.43E-01	4.80E-01	<b>5.0000e+04</b> <b>(0.0000)</b>	6.3708 (0.0240)
	MGSA	0.8400 (0.3703)	0.0172 (0.0171)	11.1163 (0.1929)	144.8417 (1.6426)	6.50E-02	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	14.6998 (0.0433)
	PNPCDE	11.2600 (0.8762)	0.0000 (0.0000)	11.3110 (0.0000)	34.1437 (8.8249)	9.53E-01	5.40E-01	<b>5.0000e+04</b> <b>(0.0000)</b>	13.5493 (0.0122)
	HTS	8.8600 (1.5780)	0.7526 (0.1332)	3.0996 (1.4566)	55.4717 (15.6015)	7.70E-01	0.00E+00	7.3565e+06 (1,287,155.0175)	177.6863 (9.6369)
	MOMMOP	11.2600 (0.6943)	0.8635 (0.0774)	1.5956 (0.8647)	25.8957 (7.0792)	9.37E-01	3.20E-01	<b>5.0000e+04</b> <b>(0.0000)</b>	20.0918 (0.2758)
	EARSDE	0.0000 (0.0000)	0.0000 (0.0000)	11.3110 (0.0000)	147.5853 (0.0000)	0.00E+00	0.00E+00	1.7467e+05 (42,697.2200)	7.1183 (0.5543)
	RM	10.1600 (1.0947)	0.6011 (0.1735)	4.5117 (1.9619)	32.1361 (12.1145)	8.45E-01	5.52E-01	<b>5.0000e+04</b> <b>(0.0000)</b>	15.1540 (1.6523)
	MCCO	<b>12.0000</b> <b>(0.0000)</b>	<b>0.9997</b> <b>(0.0002)</b>	<b>0.0036</b> <b>(0.0024)</b>	<b>1.0183</b> <b>(0.1912)</b>	<b>1.00E+00</b>	<b>1.00E+00</b>	<b>5.0000e+04</b> <b>(0.0000)</b>	4.1569 (0.5874)
$J_4$	DCGA	1.0000 (0.0000)	0.0000 (0.0000)	72.3042 (0.0000)	35.0906 (0.0000)	4.76E-02	0.00E+00	5.2699e+06 (416.2384)	4.3055 (0.1456)
	CSA	1.0000 (0.0000)	0.0000 (0.0000)	72.3042 (0.0000)	35.0906 (0.0000)	4.76E-02	0.00E+00	1.0020e+03 (0.0000)	3.5061 (0.3923)
	FSDE	1.0000 (0.0000)	0.0111 (0.0128)	71.5229 (0.8932)	34.4844 (0.6165)	4.76E-02	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	12.6042 (0.6616)
	LIPSM	16.5200 (2.0328)	3.5017 (0.7066)	212.4614 (51.9022)	12.0950 (3.6246)	7.77E-01	5.00E-01	<b>5.0000e+04</b> <b>(0.0000)</b>	<b>0.0722</b> <b>(0.0043)</b>
	LoINDE	0.0000 (0.0000)	0.0000 (0.0000)	72.3042 (0.0000)	35.0906 (0.0000)	0.00E+00	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	6.3880 (0.0158)
	MGSA	1.0000 (0.0000)	0.3235 (0.1556)	94.7625 (11.2486)	36.5793 (0.7025)	4.76E-02	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	14.9765 (0.6466)
	PNPCDE	5.2800 (1.7501)	3.1901 (1.1053)	260.7722 (64.7410)	35.2237 (1.5040)	2.29E-01	1.00E-01	<b>5.0000e+04</b> <b>(0.0000)</b>	13.5348 (0.0192)
	HTS	1.0000 (0.0000)	0.0058 (0.0071)	72.3138 (0.1896)	35.4401 (0.4779)	4.76E-02	0.00E+00	1.8194e+06 (397,438.3358)	55.4796 (7.0736)
	MOMMOP	4.1600 (3.0194)	0.4118 (0.4535)	74.1442 (15.4454)	35.5183 (1.6315)	1.83E-01	3.75E-01	<b>5.0000e+04</b> <b>(0.0000)</b>	19.8674 (1.0594)
$J_5$	EARSDE	1.0000 (0.0000)	0.0110 (0.0118)	72.5588 (0.5541)	35.7343 (0.5839)	4.76E-02	0.00E+00	2.1782e+05 (100,997.0905)	7.7488 (2.6999)
	RM	9.9600 (2.6570)	1.2543 (0.7687)	90.0817 (47.8586)	29.9317 (2.0631)	4.36E-01	4.87E-01	<b>5.0000e+04</b> <b>(0.0000)</b>	14.4105 (0.9501)
	MCCO	<b>20.2000</b> <b>(0.8367)</b>	<b>1.6587</b> <b>(0.2146)</b>	<b>58.5175</b> <b>(13.3882)</b>	<b>3.8922</b> <b>(1.7007)</b>	<b>9.62e-01</b>	<b>4.00e-01</b>	<b>5.0000e+04</b> <b>(0.0000)</b>	4.3698 (0.1319)
	DCGA	0.7800 (0.7365)	0.1181 (0.1179)	5854.3185 (782.3737)	32.5236 (5.0061)	1.56E-01	0.00E+00	5.2699e+06 (388.4161)	3.3219 (0.0074)
	CSA	0.0000 (0.0000)	0.0000 (0.0000)	6638.3536 (0.0000)	37.6246 (0.0000)	0.00E+00	0.00E+00	1.0020e+03 (0.0000)	3.2029 (0.0615)
	FSDE	2.0200 (0.1414)	0.5546 (0.0278)	2956.7455 (184.2896)	20.6989 (0.8690)	4.16E-01	2.47E-01	<b>5.0000e+04</b> <b>(0.0000)</b>	11.3648 (1.0508)
	LIPSM	4.8000 (0.4518)	0.9682 (0.0664)	210.8735 (440.7701)	1.5889 (2.6986)	9.40E-01	7.00E-01	<b>5.0000e+04</b> <b>(0.0000)</b>	<b>0.0689</b> <b>(0.0005)</b>
	LoINDE	0.0000 (0.0000)	0.0000 (0.0000)	6638.3536 (0.0000)	37.6246 (0.0000)	0.00E+00	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	6.4482 (0.0766)
	MGSA	1.0000 (0.0000)	0.0180 (0.0091)	6518.9525 (60.0834)	37.1607 (1.2430)	2.00E-01	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	14.8330 (0.1476)
PNPCDE	0.0200 (0.1414)	0.0001 (0.0010)	6637.4387 (6.4697)	37.5088 (0.8188)	4.00E-03	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	13.5464 (0.0083)	

Table 3. Cont.

Function	Algorithm	EPN	MPR	PA	DA	PR	SR	NFC	T(s)
$J_5$	HTS	3.8400 (0.9765)	0.9008 (0.1943)	1857.3834 (926.0475)	17.2088 (4.5171)	7.28E−01	8.00E−02	7.2154e+06 (1,651,796.8452)	217.8520 (32.1873)
	MOMMOP	5.0000 (0.0000)	0.8328 (0.0956)	1117.1761 (634.0152)	5.2591 (2.4087)	1.00E+00	1.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	19.1777 (0.1498)
	EARSDE	0.0000 (0.0000)	0.0000 (0.0000)	6638.3536 (0.0000)	37.6246 (0.0000)	0.00E+00	0.00E+00	6.9265e+04 (544.3150)	5.5117 (0.3721)
	RM	4.9600 (0.1979)	0.9795 (0.0404)	136.0493 (268.3073)	1.3495 (1.3138)	9.91E−01	7.25E−01	<b>5.0000e+04</b> <b>(0.0000)</b>	15.6752 (0.4225)
	MCCO	<b>5.0000</b> <b>(0.0000)</b>	<b>1.0000</b> <b>(0.0000)</b>	<b>0.1077</b> <b>(0.1451)</b>	0.0653 (0.0437)	<b>1.00e+00</b>	<b>1.00e+00</b>	<b>5.0000e+04</b> <b>(0.0000)</b>	5.1350 (0.1937)
$J_6$	DCGA	1.6800 (0.4712)	0.4852 (0.1653)	276.5939 (64.3688)	5.3019 (1.5552)	5.47E−01	1.20E−01	5.2700e+06 (389.5683)	4.6943 (0.3738)
	CSA	<b>3.0000</b> <b>(0.0000)</b>	0.0169 (0.0014)	468.6314 (0.6779)	7.7615 (0.3637)	<b>1.00E+00</b>	<b>1.00E+00</b>	1.0020e+03 (0.0000)	3.7440 (0.2425)
	FSDE	<b>3.0000</b> <b>(0.0000)</b>	0.9488 (0.0371)	24.3929 (17.6827)	0.2104 (0.1739)	<b>1.00E+00</b>	<b>1.00E+00</b>	<b>5.0000e+04</b> <b>(0.0000)</b>	11.0306 (0.3032)
	LIPSM	2.9600 (0.1979)	0.9808 (0.0686)	9.1553 (32.7096)	0.1642 (0.7120)	9.70E−01	8.34E−01	<b>5.0000e+04</b> <b>(0.0000)</b>	<b>0.0698</b> <b>(0.0033)</b>
	LoINDE	0.9400 (0.2399)	−0.0004 (0.0001)	476.9022 (0.0516)	9.0917 (0.4403)	3.33E−01	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	6.4283 (0.0049)
	MGSA	1.0000 (0.0000)	0.0025 (0.0036)	475.5096 (1.7299)	10.7513 (0.9050)	3.33E−01	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	15.2934 (0.1745)
	PNPCDE	2.9400 (0.3136)	0.0124 (0.0024)	470.7805 (1.1363)	6.9902 (0.3619)	9.87E−01	9.60E−01	<b>5.0000e+04</b> <b>(0.0000)</b>	13.5647 (0.0073)
	HTS	<b>3.0000</b> <b>(0.0000)</b>	1.0145 (0.0212)	13.2536 (4.6615)	4.2459 (0.9590)	<b>1.00E+00</b>	<b>1.00E+00</b>	4.2998e+06 (659,780.5913)	107.2854 (7.3498)
	MOMMOP	<b>3.0000</b> <b>(0.0000)</b>	0.8007 (0.0798)	95.0269 (38.0208)	2.4266 (1.5930)	<b>1.00E+00</b>	<b>1.00E+00</b>	<b>5.0000e+04</b> <b>(0.0000)</b>	19.4086 (0.2270)
	EARSDE	0.0000 (0.0000)	0.0000 (0.0000)	476.7000 (0.0000)	10.8167 (0.0000)	0.00E+00	0.00E+00	6.9417e+04 (591.6421)	5.5583 (0.3461)
$J_7$	RM	2.8600 (0.3505)	0.9164 (0.1442)	39.8426 (68.7635)	0.7597 (1.3776)	9.23E−01	8.03E−01	<b>5.0000e+04</b> <b>(0.0000)</b>	11.7645 (0.1840)
	MCCO	<b>3.0000</b> <b>(0.0000)</b>	<b>1.0000</b> <b>(0.0000)</b>	<b>0.0000</b> <b>(0.0000)</b>	<b>0.0000</b> <b>(0.0000)</b>	<b>1.00E+00</b>	<b>1.00E+00</b>	<b>5.0000e+04</b> <b>(0.0000)</b>	5.9875 (0.6458)
	DCGA	1.1800 (0.3881)	0.0995 (0.0474)	0.7495 (0.0388)	3.5085 (0.2489)	2.70E−01	2.62E−01	5.2700e+06 (423.1517)	12.9869 (0.6382)
	CSA	1.0000 (0.0000)	0.0966 (0.0543)	0.7834 (0.0452)	3.6925 (0.1322)	2.50E−01	0.00E+00	1.0020e+03 (0.0000)	3.4782 (0.4208)
	FSDE	1.0000 (0.0000)	0.1053 (0.0335)	0.7848 (0.0318)	3.7335 (0.1051)	2.50E−01	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	12.4835 (0.4654)
	LIPSM	3.8400 (0.3703)	1.1479 (0.1501)	0.1552 (0.1333)	0.6102 (0.4302)	9.95E−01	9.80E−01	<b>5.0000e+04</b> <b>(0.0000)</b>	<b>0.0707</b> <b>(0.0017)</b>
	LoINDE	1.0000 (0.0000)	0.7286 (0.0001)	1.3091 (0.0001)	4.9245 (0.0047)	2.50E−01	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	6.5180 (0.0086)
	MGSA	1.0000 (0.0000)	0.4178 (0.1458)	1.0505 (0.1213)	4.3127 (0.2471)	2.50E−01	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	14.8000 (0.2078)
	PNPCDE	1.0000 (0.0000)	0.7283 (0.0007)	1.3089 (0.0006)	4.9182 (0.0129)	2.50E−01	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	13.6422 (0.0161)
	HTS	1.0000 (0.0000)	0.0779 (0.0016)	0.7678 (0.0013)	3.6304 (0.0153)	2.50E−01	0.00E+00	2.9957e+05 (80,384.5566)	13.2584 (3.7344)
$J_8$	MOMMOP	1.0000 (0.0000)	0.0859 (0.0099)	0.7745 (0.0083)	3.6868 (0.0452)	2.50E−01	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	20.6801 (1.8592)
	EARSDE	1.0000 (0.0000)	0.0775 (0.0000)	0.7675 (0.0000)	3.6242 (0.0002)	2.50E−01	0.00E+00	2.0720e+05 (99,520.1634)	9.7879 (1.7101)
	RM	1.4800 (0.5047)	0.3347 (0.1999)	0.7724 (0.1356)	4.1215 (0.5835)	3.75E−01	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	12.2837 (0.5762)
	MCCO	<b>4.0000</b> <b>(0.0000)</b>	<b>1.1222</b> <b>(0.0417)</b>	<b>0.1017</b> <b>(0.0347)</b>	<b>0.3258</b> <b>(0.1017)</b>	<b>1.00E+00</b>	<b>1.00E+00</b>	<b>5.0000e+04</b> <b>(0.0000)</b>	7.1484 (0.6582)

Table 4. Numerical results for  $J_8$ – $J_{14}$  multimodal test functions.

Function	Algorithm	EPN	MPR	PA	DA	PR	SR	NFC	T(s)
$J_8$	DCGA	6.7400 (0.8992)	0.5091 (0.1381)	57.9157 (16.2795)	2.1725 (0.9493)	8.60E−01	3.00E−01	5.2700e+06 (423.5090)	3.9790 (0.0227)
	CSA	1.0000 (0.0000)	0.1357 (0.0000)	101.9679 (0.0000)	8.2522 (0.0000)	1.25E−01	0.00E+00	1.0020e+03 (0.0000)	3.6211 (0.4123)
	FSDE	7.8200 (0.4819)	0.9358 (0.0614)	7.5921 (7.2334)	0.4674 (0.6016)	9.85E−01	8.80E−01	<b>5.0000e+04</b> <b>(0.0000)</b>	11.0149 (0.2440)
	LIPSM	6.3600 (1.3815)	0.2437 (0.4316)	89.1749 (50.8793)	2.7790 (1.7139)	7.58E−01	1.40E−01	<b>5.0000e+04</b> <b>(0.0000)</b>	<b>0.0705</b> <b>(0.0005)</b>
	LoINDE	1.0000 (0.0000)	−0.4791 (0.0002)	174.3908 (0.0191)	8.9533 (0.0077)	1.25E−01	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	6.3788 (0.0431)
	MGSA	1.0000 (0.0000)	−0.0566 (0.1507)	124.5808 (17.7644)	8.5178 (0.1649)	1.25E−01	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	14.5061 (0.5020)
	PNPCDE	1.0000 (0.0000)	−0.4702 (0.0119)	173.3371 (1.4049)	8.9644 (0.0348)	1.25E−01	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	13.5234 (0.0121)
	HTS	1.0000 (0.0000)	0.1345 (0.0016)	102.0548 (0.1845)	8.2801 (0.1379)	1.25E−01	0.00E+00	7.8946e+05 (237,933.4253)	19.1952 (7.6346)
	MOMMOP	1.0000 (0.0000)	0.1296 (0.0040)	102.6255 (0.4688)	8.7652 (0.5011)	1.25E−01	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	22.3204 (0.8625)
	EARSDE	1.0000 (0.0000)	0.1350 (0.0029)	102.0438 (0.3251)	8.3202 (0.2762)	1.25E−01	0.00E+00	2.3699e+05 (105,559.4854)	7.5346 (1.5782)
	RM	1.6600 (1.3494)	0.1324 (0.1502)	102.2930 (17.7047)	8.5933 (0.4921)	2.47E−01	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	13.1458 (0.0762)
	MCCO	<b>8.0000</b> <b>(0.0000)</b>	<b>0.8051</b> <b>(0.1250)</b>	<b>23.0527</b> <b>(14.7310)</b>	<b>0.5634</b> <b>(0.2429)</b>	<b>1.00E+00</b>	<b>1.00E+00</b>	<b>5.0000e+04</b> <b>(0.0000)</b>	4.7895 (0.3612)
$J_9$	DCGA	5.1000 (0.6468)	0.5901 (0.0878)	2.3783 (0.4837)	1.7795 (0.5792)	8.67E−01	3.40E−01	5.2699e+06 (416.2275)	6.3541 (0.3430)
	CSA	1.0000 (0.0000)	0.1713 (0.0147)	4.7577 (0.0660)	6.2613 (0.6894)	1.67E−01	0.00E+00	1.0020e+03 (0.0000)	3.4052 (0.2621)
	FSDE	5.8000 (0.4041)	0.8141 (0.0852)	1.0691 (0.4780)	0.5166 (0.4258)	9.53E−01	7.20E−01	<b>5.0000e+04</b> <b>(0.0000)</b>	11.4995 (0.4801)
	LIPSM	4.4800 (0.8862)	0.4685 (0.1213)	3.0125 (0.6852)	2.4932 (0.8558)	7.33E−01	8.00E−02	<b>5.0000e+04</b> <b>(0.0000)</b>	<b>0.0696</b> <b>(0.0013)</b>
	LoINDE	4.5000 (0.5051)	0.0016 (0.0002)	5.6544 (0.0010)	11.8864 (0.6904)	7.13E−01	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	5.9209 (0.0470)
	MGSA	1.0000 (0.0000)	0.0798 (0.0265)	5.2112 (0.1500)	6.3464 (0.5928)	1.67E−01	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	14.8454 (0.1165)
	PNPCDE	4.1800 (0.3881)	0.0015 (0.0001)	5.6548 (0.0008)	11.4256 (0.5542)	7.07E−01	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	13.7502 (0.2281)
	HTS	1.0000 (0.0000)	0.1752 (0.0017)	4.7406 (0.0082)	6.3731 (0.5063)	1.67E−01	0.00E+00	1.9116e+06 (292,922.6258)	46.6750 (0.1455)
	MOMMOP	1.0000 (0.0000)	0.1423 (0.0146)	4.8574 (0.0825)	6.2275 (0.6006)	1.67E−01	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	20.2234 (0.1947)
	EARSDE	1.0000 (0.0000)	0.1765 (0.0002)	4.7471 (0.0009)	6.2540 (0.6985)	1.67E−01	0.00E+00	1.9827e+05 (104,111.6689)	7.2288 (0.9528)
	RM	1.1600 (0.3703)	0.1375 (0.0513)	4.8846 (0.2907)	6.1834 (0.7370)	1.95E−01	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	11.8541 (0.0930)
	MCCO	<b>6.0000</b> <b>(0.0000)</b>	<b>0.9942</b> <b>(0.0585)</b>	<b>0.3756</b> <b>(0.3233)</b>	<b>0.1208</b> <b>(0.0878)</b>	<b>1.00E+00</b>	<b>1.00E+00</b>	<b>5.0000e+04</b> <b>(0.0000)</b>	5.5784 (0.3712)
$J_{10}$	DCGA	17.3800 (4.1349)	0.2295 (0.0752)	53.7829 (5.2159)	81.5710 (15.9520)	4.89E−01	0.00E+00	5.2699e+06 (406.6097)	4.4611 (0.3421)
	CSA	1.6400 (1.3962)	1.2028 (1.9805)	31.2207 (1.8519)	157.2020 (0.5684)	7.78E−03	0.00E+00	1.0020e+03 (0.0000)	3.7716 (0.4670)
	FSDE	5.5800 (2.0711)	0.1564 (0.0552)	58.8156 (3.8439)	114.5919 (10.6299)	1.58E−01	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	11.0075 (0.2962)
	LIPSM	18.5000 (6.9818)	0.1415 (0.1798)	59.9756 (12.3843)	102.5510 (32.2349)	5.14E−01	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	<b>0.0719</b> <b>(0.0026)</b>
	LoINDE	31.0400 (5.5401)	−0.8857 (0.1576)	131.4681 (10.9899)	82.3903 (13.1937)	8.14E−01	8.00E−02	<b>5.0000e+04</b> <b>(0.0000)</b>	6.3637 (0.0051)

Table 4. Cont.

Function	Algorithm	EPN	MPR	PA	DA	PR	SR	NFC	T(s)
$J_{10}$	MGSA	1.0000 (0.0000)	-0.0025 (0.0147)	69.8919 (1.0264)	154.1091 (1.0567)	2.78E-02	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	14.4385 (0.1911)
	PNPCDE	<b>34.5200</b> <b>(2.3408)</b>	-0.9657 (0.0670)	137.0452 (4.6695)	80.5974 (5.9311)	<b>9.62E-01</b>	<b>4.00E-02</b>	<b>5.0000e+04</b> <b>(0.0000)</b>	13.4963 (0.0070)
	HTS	22.6400 (8.8865)	0.0025 (0.0017)	69.5456 (0.1159)	111.9837 (18.2089)	6.14E-01	4.00E-02	4.7138e+06 (482,108.0580)	88.7629 (3.5432)
	MOMMOP	27.0000 (6.8243)	0.5971 (0.1727)	28.5716 (11.5581)	70.6105 (8.6683)	8.79E-01	1.60E-01	<b>5.0000e+04</b> <b>(0.0000)</b>	20.8947 (0.8648)
	EARSDE	0.4400 (1.8534)	1.4477 (10.4475)	39.4683 (11.8950)	156.7319 (3.1499)	7.78E-03	0.00E+00	2.1237e+05 (118,498.9689)	7.8712 (1.3295)
	RM	23.3400 (10.6342)	0.2915 (0.3093)	49.8345 (20.9848)	99.8005 (35.7264)	5.61E-03	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	13.7253 (0.6319)
	MCCO	25.0000 (0.0000)	0.7064 (0.0127)	<b>21.1987</b> <b>(0.7800)</b>	<b>35.8584</b> <b>(1.1774)</b>	6.94e-01	0.00e+00	<b>5.0000e+04</b> <b>(0.0000)</b>	4.1574 (0.3804)
$J_{11}$	DCGA	32.5400 (2.0723)	0.2892 (0.1015)	50.3347 (6.7100)	25.6746 (3.7450)	8.26E-01	2.00E-02	5.2702e+06 (452.3383)	4.3025 (0.4210)
	CSA	14.7600 (10.3541)	0.1603 (0.1122)	58.2375 (7.7804)	79.9445 (4.5951)	2.85E-01	0.00E+00	1.0020e+03 (0.0000)	3.8122 (0.8740)
	FSDE	<b>33.3600</b> <b>(1.4394)</b>	<b>0.9261</b> <b>(0.0387)</b>	<b>20.9214</b> <b>(1.8189)</b>	25.7048 (2.3848)	<b>8.26E-01</b>	<b>0.00E+00</b>	<b>5.0000e+04</b> <b>(0.0000)</b>	11.3063 (0.7670)
	LIPSM	33.2800 (2.8287)	-0.0845 (0.2927)	75.7633 (19.7202)	<b>24.0068</b> <b>(4.8334)</b>	8.51E-01	2.00E-02	<b>5.0000e+04</b> <b>(0.0000)</b>	<b>0.0701</b> <b>(0.0013)</b>
	LoINDE	30.8600 (5.7924)	-1.8075 (0.3318)	194.7076 (23.0088)	54.7586 (3.9624)	7.76E-01	4.00E-02	<b>5.0000e+04</b> <b>(0.0000)</b>	6.3762 (0.0169)
	MGSA	1.0000 (0.0000)	-0.0128 (0.0084)	70.2408 (0.5795)	81.8813 (0.4700)	2.50E-02	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	14.0801 (0.0271)
	PNPCDE	16.7400 (10.8624)	-0.2905 (0.1866)	89.5021 (12.9423)	77.3101 (5.0775)	3.99E-01	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	13.4828 (0.0089)
	HTS	23.3400 (11.3616)	1.6178 (0.9708)	98.8315 (34.8418)	121.6556 (25.5682)	5.95E-01	4.00E-02	5.3787e+06 (1,538,699.7330)	114.3261 (40.2107)
	MOMMOP	29.9400 (6.6253)	0.8442 (0.2780)	37.8543 (7.4293)	65.4402 (4.8254)	7.03E-01	1.20E-01	<b>5.0000e+04</b> <b>(0.0000)</b>	18.9512 (0.5326)
	EARSDE	0.0000 (0.0000)	0.0000 (0.0000)	69.3526 (0.0000)	81.7565 (0.0000)	0.00E+00	0.00E+00	2.2695e+05 (22,116.5083)	7.7541 (0.5658)
	RM	32.5000 (4.2964)	0.1675 (0.3229)	59.8485 (19.3763)	51.7364 (6.1193)	8.16E-01	4.27E-02	<b>5.0000e+04</b> <b>(0.0000)</b>	12.1458 (0.3897)
MCCO	24.6000 (1.1402)	0.3816 (0.0327)	43.1790 (1.9530)	37.2566 (3.2108)	6.15e-01	0.00e+00	<b>5.0000e+04</b> <b>(0.0000)</b>	4.5769 (0.1642)	
$J_{12}$	DCGA	3.5600 (1.2149)	0.4143 (0.1232)	18.9560 (3.9532)	44.4814 (12.2185)	4.68E-01	0.00E+00	5.2700e+06 (512.5000)	4.4157 (0.1919)
	CSA	0.0000 (0.0000)	0.0000 (0.0000)	32.1222 (0.0000)	74.7141 (0.0000)	8.75E-02	0.00E+00	1.0020e+03 (0.0000)	3.5020 (0.0492)
	FSDE	2.0000 (0.0000)	0.3800 (0.0012)	19.9148 (0.0371)	56.3252 (0.0584)	2.50E-01	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	11.5497 (0.5992)
	LIPSM	3.6000 (1.1429)	0.4405 (0.1499)	17.9719 (4.8161)	48.2005 (10.5583)	4.55E-01	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	0.0692 (0.0007)
	LoINDE	6.4800 (1.0349)	-0.8734 (0.1161)	60.1790 (3.7304)	31.5892 (6.1687)	7.70E-01	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	6.4375 (0.0281)
	MGSA	1.0000 (0.0000)	-0.0262 (0.0667)	32.9630 (2.1425)	68.4868 (1.3019)	1.25E-01	0.00E+00	<b>5.0000e+04</b> <b>(0.0000)</b>	14.0832 (0.0301)
	PNPCDE	7.1000 (0.3030)	-0.6212 (0.1318)	52.0763 (4.2339)	26.1353 (2.7162)	8.88E-01	1.00E-01	<b>5.0000e+04</b> <b>(0.0000)</b>	13.5545 (0.0269)
	HTS	4.5400 (1.1988)	0.4554 (0.3582)	26.5703 (4.0784)	47.7845 (8.3195)	5.40E-01	0.00E+00	3.4108e+06 (546,220.0657)	83.3420 (18.7297)
	MOMMOP	7.7800 (0.4647)	0.8766 (0.1026)	5.8293 (2.7242)	10.1018 (4.9705)	9.75E-01	8.00E-01	<b>5.0000e+04</b> <b>(0.0000)</b>	19.0711 (0.7056)
	EARSDE	0.8200 (0.3881)	0.0900 (0.1135)	30.0505 (2.7604)	70.7956 (3.0980)	8.75E-02	0.00E+00	3.4856e+05 (128,156.0640)	11.1211 (1.9141)

Table 4. Cont.

Function	Algorithm	EPN	MPR	PA	DA	PR	SR	NFC	T(s)
$J_{12}$	RM	4.5200 (0.6465)	0.4752 (0.0674)	16.8572 (2.1657)	42.4463 (6.8784)	5.87E-01	0.00E+00	<b>5.0000e+04</b> (0.0000)	11.6824 (0.6375)
	MCCO	<b>8.0000</b> (0.0000)	<b>0.9997</b> (0.0002)	<b>0.0159</b> (0.0074)	<b>0.2078</b> (0.0242)	<b>1.00E+00</b>	<b>1.00E+00</b>	<b>5.0000e+04</b> (0.0000)	4.2497 (0.3921)
	DCGA	1.0000 (0.0000)	-0.1474 (0.0011)	<b>1.7324</b> (0.0008)	5.7077 (0.1723)	8.33E-02	0.00E+00	5.2700e+06 (396.7548)	4.0571 (0.2988)
	CSA	1.0000 (0.0000)	-0.1337 (0.0295)	1.7442 (0.0249)	5.7767 (0.1971)	8.33E-02	0.00E+00	1.0020e+03 (0.0000)	3.1997 (0.0902)
	FSDE	1.0000 (0.0000)	-0.1462 (0.0024)	1.7332 (0.0020)	5.6966 (0.1770)	8.33E-02	0.00E+00	<b>5.0000e+04</b> (0.0000)	11.5354 (0.2906)
	LIPSM	1.0800 (0.2740)	-0.0513 (0.2371)	1.7869 (0.1180)	5.7190 (0.1694)	8.67E-02	0.00E+00	<b>5.0000e+04</b> (0.0000)	0.0679 (0.0007)
	LoINDE	1.0000 (0.0000)	2.0893 (0.0181)	3.6384 (0.0155)	6.8529 (0.1807)	8.33E-02	0.00E+00	<b>5.0000e+04</b> (0.0000)	5.8972 (0.0177)
$J_{13}$	MGSA	1.0000 (0.0000)	0.1255 (0.1769)	1.9647 (0.1507)	5.8757 (0.1979)	8.33E-02	0.00E+00	<b>5.0000e+04</b> (0.0000)	14.7684 (0.0800)
	PNPCDE	1.0000 (0.0000)	1.7767 (0.3247)	3.3719 (0.2768)	6.7406 (0.1949)	8.33E-02	0.00E+00	<b>5.0000e+04</b> (0.0000)	13.5227 (0.0248)
	HTS	1.0000 (0.0000)	-0.1479 (0.0000)	<b>1.7324</b> (0.0000)	5.7680 (0.1847)	8.33E-02	0.00E+00	4.2065e+05 (67,438.1820)	12.4227 (0.2386)
	MOMMOP	1.0000 (0.0000)	-0.1310 (0.0224)	1.7460 (0.0191)	5.7757 (0.1697)	8.33E-02	0.00E+00	<b>5.0000e+04</b> (0.0000)	18.8628 (0.3494)
	EARSDE	1.0000 (0.0000)	-0.1479 (0.0000)	1.7324 (0.0000)	5.7823 (0.1738)	8.33E-02	0.00E+00	2.4215e+05 (104,975.1893)	7.8763 (1.4696)
	RM	1.0000 (0.0000)	-0.0915 (0.0558)	1.7797 (0.0475)	5.8308 (0.1042)	8.33E-02	0.00E+00	<b>5.0000e+04</b> (0.0000)	12.2543 (0.4314)
	MCCO	<b>4.8000</b> (0.4472)	-0.6356 (0.0321)	1.8985 (0.0279)	<b>5.3181</b> (0.1704)	<b>4.00e-01</b>	<b>0.00e+00</b>	<b>5.0000e+04</b> (0.0000)	4.4547 (0.4962)
$J_{14}$	DCGA	1.0000 (0.0000)	0.0000 (0.0000)	114.3627 (0.0000)	28.8892 (0.0000)	1.11E-01	0.00E+00	5.2702e+06 (408.7596)	3.6593 (0.0113)
	CSA	1.0000 (0.0000)	0.0000 (0.0000)	114.3627 (0.0000)	28.8892 (0.0000)	1.11E-01	0.00E+00	1.0020e+03 (0.0000)	3.4427 (0.3641)
	FSDE	1.0000 (0.0000)	0.0001 (0.0002)	114.3570 (0.0160)	28.8916 (0.0126)	1.11E-01	0.00E+00	<b>5.0000e+04</b> (0.0000)	12.3678 (0.1444)
	LIPSM	5.4400 (1.8201)	0.6144 (0.2655)	65.6156 (23.4835)	14.7539 (6.0802)	5.69E-01	4.00E-02	<b>5.0000e+04</b> (0.0000)	<b>0.0677</b> (0.0005)
	LoINDE	4.0400 (0.1979)	3.3509 (0.0956)	348.3661 (6.7529)	23.1722 (0.8612)	4.44E-01	0.00E+00	<b>5.0000e+04</b> (0.0000)	6.3759 (0.0272)
	MGSA	1.0000 (0.0000)	0.2210 (0.1169)	139.5238 (13.3732)	30.2407 (0.7448)	1.11E-01	0.00E+00	<b>5.0000e+04</b> (0.0000)	14.6457 (0.0936)
	PNPCDE	6.0000 (0.0000)	2.8921 (0.0033)	292.6285 (0.3751)	22.4646 (0.0682)	6.67E-01	0.00E+00	<b>5.0000e+04</b> (0.0000)	13.5294 (0.0155)
$J_{15}$	HTS	1.2400 (0.8221)	0.0276 (0.0916)	112.8990 (5.3421)	28.7942 (0.6401)	1.24E-01	0.00E+00	2.5309e+06 (674,403.9839)	83.0052 (10.2085)
	MOMMOP	6.0800 (0.8291)	0.7335 (0.1540)	54.9871 (7.7088)	19.7481 (2.6027)	7.16E-01	0.00E+00	<b>5.0000e+04</b> (0.0000)	18.9710 (0.3317)
	EARSDE	1.0000 (0.0000)	0.0000 (0.0000)	114.3627 (0.0000)	28.8892 (0.0001)	1.11E-01	0.00E+00	2.0328e+05 (107,472.8440)	8.7347 (2.4919)
	RM	7.3000 (1.0351)	1.3945 (0.4019)	105.6598 (30.2145)	13.0871 (3.4396)	8.01E-01	0.00E+00	<b>5.0000e+04</b> (0.0000)	11.6824 (0.6375)
	MCCO	<b>9.0000</b> (0.0000)	<b>0.9323</b> (0.1354)	<b>0.4787</b> (0.0089)	<b>0.4281</b> (0.0235)	<b>1.00E+00</b>	<b>1.00E+00</b>	<b>5.0000e+04</b> (0.0000)	4.0872 (0.2458)

Table 3 reports the numerical results for functions  $J_1$ – $J_7$ . In the table, it can be seen that MCCO and MOMMOP achieved the best results for the majority of the numerical simulations. In function  $J_1$  and  $J_5$ , both algorithms outperformed the others. According to the EPN metric, only MCCO and MOMMOP are capable of finding the total number of peaks in the functions; however, MCCO produced

more consistent results. In the case of function  $J_2$ , FSDE and MCCO obtained all the optimal values. For function  $J_6$ , CSA, HTS, FSDE MOMMOP, and MCCO obtained the maximum number of peaks with similar levels of robustness. The most distinguishing characteristics of MCCO regarding these multimodal methods concerned its ability to find all the peaks with relatively low computational effort, compared to its competitors. For functions  $J_3$ ,  $J_4$  and  $J_7$ , it is clear that MCCO outperformed the other algorithms in terms of dispersity, scalability, and precision. According to the reported results, the MPR, PA, and DA metrics suggest that MCCO is capable of operating under complex multimodal functions by yielding the greatest EPN value.

Additionally, Table 3 reports the computational effort of each multimodal approach, considering the NFC and execution time. From the results, it is quite evident that LIPSM presented the best execution time metric for functions  $J_1$ – $J_7$ . However, performance metrics indicate that LIPSM was not able to produce competitive results. In contrast, MCCO yielded significantly better results than many of the tested algorithms when evaluating a similar number of function calls. Considering the execution time, it can be seen that MCCO presented better results than DCGA, FSDE, LoINDE, MGSA, PNPCE, HTS, MOMMOP, EARSDE, and RM in function  $J_1$ . For function  $J_2$ ,  $J_4$ , and  $J_5$ , MCCO outperformed FSDE, LoINDE, MGSA, PNPCE, HTS, MOMMOP, EARSDE, and RM. Also MCO beat DCGA, CSA, FSDE, LoINDE, MGSA, PNPCE, HTS, MOMMOP, EARSDE, and RM on function  $J_3$ . For functions  $J_6$  and  $J_7$ , MCCO also presented remarkable results.

In general terms, by analyzing the numerical results from Table 3, it can be seen that the competitive memory approach adapted in the original operators of the CCO method provided better results than its competitors. Since collective memory is based on the concept of dominance, potential solutions compete among themselves to be allocated into the historic memory. This process detects most of the possible optima in a single run of the entire MCCO method. The DA metric reported that MCCO obtained the optimal values with the shortest spatial relation, compared to the rest of multimodal methods. This indicates that MCCO produces solutions with a higher level of consistency. The MPR and PA performance metrics corroborate that the proposed approach obtained a higher accuracy level than the other methodologies by measuring the error among approximated optima and true optimal values, evaluating each benchmark function in fewer runs with respect to the other methods. This indicates that the proposed mechanism is capable of finding most of the true optima with low computational effort.

From the numerical results in Table 4, it is clear that for functions  $J_{10}$  and  $J_{11}$ , PNPCE and FSDE outperformed the other algorithms, including MCCO. The experimental results suggest that these methods are capable of finding a higher number of optimal values than MCCO. However, PNPCE presented a lower level of consistency than MCCO. Considering the standard deviation of EPN, it can be seen that PNPCE and FSDE tended to produce dispersed solutions. Under such circumstances, both techniques produced nonrobust solutions. In contrast, even though it did not detect the highest number of peaks in functions  $J_{10}$  and  $J_{11}$ , the standard deviation of EPN indicated that more consistent and robust solutions were produced using MCCO. On the other hand, for the remaining functions, MCCO outperformed the tested multimodal approaches. In functions  $J_{08}$  and  $J_{09}$ , it was quite evident that the competition phase from the memory mechanism in CCO efficiently registered most of the candidate optima for these functions. In the results, it can be noted that MCCO produced results containing a higher level of accuracy with the lowest standard deviations. Additionally, for functions  $J_{12}$ – $J_{14}$ , MCCO presented remarkable performance considering the EPN and its corresponding standard deviation. Also, MCCO produced closer solutions regarding the true optima for each benchmark function (MPR). MCCO is capable of detecting the suboptimal and optimal solution under these functions, since it makes use of a powerful updating scheme. The collective memory mechanism stores and manages all the potential solutions thanks to the balance among the original evolutionary operators of the CCO method.

Considering the computational effort metrics, MCCO requires similar function calls than FSDE, LIPSM, LoINDE, MGSA, PNPCE, MOMMOP, and RM. However, MCCO presents better execution

time metrics in functions  $J_8, J_{11}, J_{13}$ , and  $J_{14}$  than FSDE, LoINDE, PNPCE, HTS, MOMMOP, EARSDE, and RM; additionally, for functions  $J_9, J_{10}$ , and  $J_{12}$ , it outperforms DCGA, FSDE, LoINDE, MGSA, PNPCE, HTS, MOMMOP, EARSDE, and RM.

As a result, the proposed multimodal extension to adapt CCO in multimodal optimization produces a balanced and powerful data structure which can efficiently register, maintain, and manage all potential solutions during the entire optimization process. Also, from the experimental study, it was determined that the proposed MCCO detects most of the optima for the majority of the test functions, evaluating similar NFC than the many of tested methods with the lowest execution time in most cases, indicating that MCCO is less computationally complex.

In order to statistically corroborate the numerical results from Tables 3 and 4, A Wilcoxon rank sum test was computed. This nonparametric approach indicates whether there is a significant difference between two multimodal approaches. In this study, the test was conducted based on 5% significance. Table 5 report the  $p$ -values of a pair-wise comparison among each multimodal technique. For the test, the proposed null hypothesis  $H_0$  represents the idea of no significant difference among multimodal methods. As a counterpart, the proposed alternative hypothesis  $H_1$  indicates a significant difference among two tested methods. To visually analyze the numerical results from this test, Table 5 uses the symbols ▲, ▼, and ►; the first symbol indicates that one algorithm outperformed its competitor; the second indicates that a given method performed worse than its competitor; the third symbol indicates that the statistical test could not decide which algorithm was significantly better. As shown in Table 5, MCCO performed better than its competitors, producing solutions that were quite different in most of the experimental cases.

**Table 5.**  $p$ -values produced by nonparametric test comparing MCCO vs. DCGA, MCCO vs. MOCO, MOCO vs. FSDE, MCCO vs. LIPSM, MCCO vs. LoINDE, MCCO vs. MGSA, MCCO vs. PNPCE, MCCO vs. HTS, MCCO vs. MOMMOP, MCCO vs. EARSDE, and MCCO vs. RM over the EPN performance metric from Tables 2 and 3.

MCCO vs.	DCGA	CSA	FSDE	LIPSM	LoINDE	MGSA	PNPCDE	HTS	MOMMOP	EARS DE	RM
$J_1$	1.47E-13▲	8.01E-05▲	1.35E-04▲	2.78E-13▲	2.69E-13▲	2.69E-13▲	1.22E-05▲	3.97E-07▲	0.00E+00►	2.69E-13▲	1.14E-04▲
$J_2$	1.41E-04▲	2.50E-13▲	0.00E+00►	5.80E-07▲	3.26E-03▲	2.50E-13▲	9.48E-07▲	1.29E-04▲	1.15E-04▲	2.50E-13▲	7.47E-06▲
$J_3$	2.14E-04▲	2.66E-13▲	1.33E-03▲	5.23E-06▲	1.01E-04▲	5.81E-08▲	7.40E-05▲	1.34E-04▲	5.69E-05▲	2.66E-13▲	2.87E-10▲
$J_4$	3.08E-10▲	2.69E-13▲	2.69E-13▲	3.36E-11▲	2.69E-13▲	2.69E-13▲	1.18E-05▲	2.69E-13▲	1.48E-04▲	2.69E-13▲	4.48E-09▲
$J_5$	1.65E-04▲	2.65E-13▲	1.50E-08▲	1.33E-08▲	2.65E-13▲	2.65E-13▲	1.62E-11▲	1.20E-09▲	0.00E+00►	2.65E-13▲	1.87E-07▲
$J_6$	1.43E-05▲	0.00E+00►	0.00E+00►	8.00E-06▲	2.50E-13▲	2.50E-13▲	6.85E-08▲	0.00E+00►	0.00E+00►	2.50E-13▲	6.87E-07▲
$J_7$	1.71E-07▲	2.50E-13▲	2.50E-13▲	4.32E-08▲	2.50E-13▲	2.50E-13▲	2.50E-13▲	2.50E-13▲	2.50E-13▲	2.50E-13▲	5.24E-14▲
$J_8$	5.21E-03▲	2.50E-13▲	4.32E-01▲	1.28E-09▲	2.50E-13▲	2.50E-13▲	2.50E-13▲	2.50E-13▲	2.50E-13▲	2.50E-13▲	8.45E-18▲
$J_9$	3.31E-03▲	2.50E-13▲	1.81E-01▲	5.92E-08▲	1.37E-05▲	2.50E-13▲	7.15E-06▲	2.50E-13▲	2.50E-13▲	2.50E-13▲	7.13E-11▲
$J_{10}$	5.94E-03▲	4.54E-07▲	1.33E-04▲	2.02E-04▲	1.24E-02▼	2.68E-13▲	5.33E-06▼	7.24E-08▲	9.32E-04▼	3.20E-05▲	9.67E-08▲
$J_{11}$	2.44E-04▲	2.27E-03▲	2.33E-04▲	2.43E-04▼	1.59E-02▼	2.69E-13▲	7.98E-07▲	3.95E-07▲	4.06E-02▼	2.69E-13▲	2.58E-07▼
$J_{12}$	2.80E-04▲	2.65E-13▲	2.65E-13▲	3.54E-04▲	6.40E-08▲	2.65E-13▲	1.45E-04▲	8.08E-04▲	2.85E-05▲	5.91E-05▲	7.59E-13▲
$J_{13}$	2.68E-13▲	2.68E-13▲	2.68E-13▲	2.77E-09▲	2.68E-13▲	2.68E-13▲	2.68E-13▲	2.68E-13▲	2.68E-13▲	2.68E-13▲	1.57E-17▲
$J_{14}$	2.66E-13▲	2.66E-13▲	2.66E-13▲	2.68E-02▲	2.66E-13▲	2.66E-13▲	7.89E-06▲	2.74E-09▲	1.33E-08▲	2.66E-13▲	6.31E-18▲
▲	14	13	12	13	12	14	13	13	9	14	13
▼	0	0	0	1	2	0	1	0	2	0	1
►	0	1	2	0	0	0	0	1	3	0	0

### 5. Conclusions

Evolutionary Computation Methods (ECMs) are stochastic search mechanisms which present an alternative search strategy with which to solve real-world optimization problems where classical optimization techniques are unsuitable. Most of the literature on ECMs indicates that these methods are conceived of to detect the global optimum. However, in real-world applications in the engineering, medical, or economic fields, the global optimum may not be realizable due to physical, mechanical,

or even realistic aspects. Under such circumstances, multimodal optimization methodologies have been designed to detect optimal and suboptimal solutions in a single run of the optimization process.

This paper presents a multimodal extension to incorporate multimodal capabilities in a recently developed optimization algorithm called Cluster-Chaotic-Optimization (CCO). The proposed Multimodal Cluster-Chaotic-Optimization (MCCO) incorporates the concept of dominance found in animal behavior, which indicates the level of social interaction between two animals in terms of aggressiveness. Such aggressiveness leads the animals to remain as distant as possible from each other, i.e., the most dominant individual prevails while the other withdraws. In MCCO, this concept is computationally abstracted in terms of a data structure called “competitive memory” to incorporate multimodal capabilities into the evolutionary operators of the CCO.

The single optimum CCO divides the population into small clusters for each generation; meanwhile, the search strategy is conducted based on intra- and extra- cluster evolutionary operations. Intracuster procedures will cause the search strategy to be inside each cluster. Extracuster will search outside of each cluster but inside the feasible search space. The combination of these two evolutionary operators tends to form groups into potential search zones. Such promissory zones can be efficiently registered within a memory data structure to maintain potential locations which can be catalogued as optimal and suboptimal values. Under such circumstances, CCO can be extended considering the abstraction idea of animal dominance to incorporate multimodal capabilities into the original CCO to detect all possible optimal solutions in a single run of the optimization process.

The performance of the proposed MCCO was tested and compared against eleven multimodal techniques, i.e., DCGA, CSA, FSDE, LIPSM, LoINDE, MGSA, PNPCDE, HTS, MOMMOP, EARSDE, and RM. In the experimental section, a comparison of the results based on six commonly used multimodal performance metrics, i.e., Effective Peak Number (EPN), the Maximum Peak Ratio (MPR), the Peak Accuracy (PA), the Distance Accuracy (DA), the Peak Ratio (PR), and the Success Rate (SR) was reported. The EPN reflects the ability of a multimodal technique to obtain most of the optima. The MPR computes the consistency of the approximated optima over true optima. On the other hand, PA measures the error among approximated optima and true optima. Similarly, DA indicates the total error, considering each independent variable of the objective function. PR calculates the percentage of the total number of approximated optima over multiple executions of a given algorithm. Lastly, SR measures the successful percentage of runs, considering the total number of executions. Also, the computational effort, in terms of the Number of Function Calls (NFC) and execution time (T) in seconds, was reported.

Based on the numerical results, it was demonstrated that the proposed approach is capable of obtaining most of the true optimal values in most of the benchmark functions, with a competitive computational effort level based on NFC and execution time. Since the MPR, PA, DA, PR, and SR metrics were calculated based on the EPN metric, a nonparametric test was conducted on the EPN metric to statistically validate the performance results based on true optima approximation. In the statistical test, it was shown that the proposed method is capable of locating most of the true optima based on the Euclidean distance between the true optima and approximated solutions.

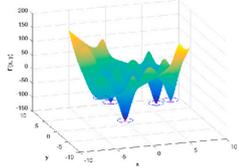
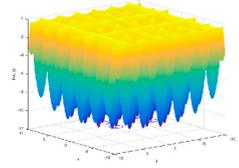
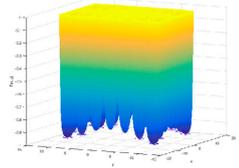
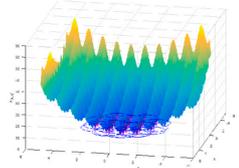
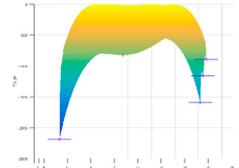
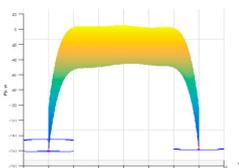
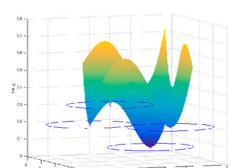
**Author Contributions:** Formal analysis, J.G.; Methodology, E.C.; Software, J.G. and K.G.D.; Writing—original draft, J.G. and E.C.; Writing—review & editing, K.G.D. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

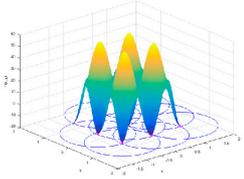
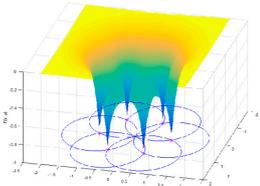
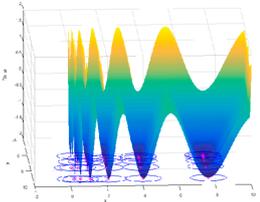
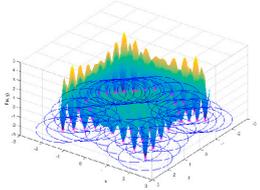
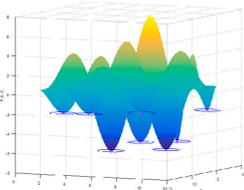
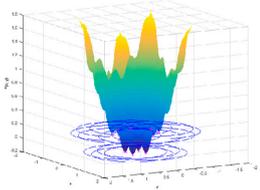
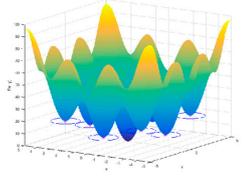
**Conflicts of Interest:** On behalf of all authors, the corresponding author states that there is no conflict of interest. The authors declare that they have no conflict of interest.

Appendix A Multimodal Test Functions Formulation

Table A1. Multimodal test functions  $J_1$ – $J_7$  used in the experimental study.

Function	Search Domain	Dimensionality	Optima Number	Graph
<b>Bird</b>				
$J_1(x_1, x_2) = \sin(x_1) \cdot e^{(1 - \cos(x_2))^2} + \cos(x_2) \cdot e^{(1 - \sin(x_1))^2} + (x_1 - x_2)^2$	$[-2\pi, 2\pi]^n$	$n = 2$	6	
<b>Test Tube Holder</b>				
$J_2(x_1, x_2) = -4 \cdot \left  e^{\cos\left(\frac{x_1^2 + x_2^2}{200}\right)} \cdot \sin(x_1) \cdot \cos(x_2) \right $	$[-10, 10]^n$	$n = 2$	4	
<b>Penholder</b>				
$J_3(x_1, x_2) = -e^{e^{\frac{\sqrt{x_1^2 + x_2^2}}{4} + 11} \cdot \cos(x_1) \cdot \cos(x_2)}$	$[-11, 11]^n$	$n = 2$	12	
<b>Rastrigin</b>				
$J_4(x_1, x_2) = 10n \cdot \sum_{i=1}^n x_i^2 - 10 \cdot \cos(2\pi x_i)$	$[-5.12, 5.12]^n$	$n = 2$	21	
<b>Himmelblau</b>				
$J_5(x_1, x_2) = -\left( (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 \right)$	$[-6, 6]^n$	$n = 2$	5	
<b>Six Hump Camel</b>				
$J_6(x_1, x_2) = -(4x_1^2 + x_1x_2 - 4x_2^2 - 2.1x_1^4 + 4x_2^4 + \frac{1}{3}x_1^6)$	$x_1 = [-3, 3]$ $x_2 = [-2, 2]$	$n = 2$	3	
<b>Giunta</b>				
$J_7(x_1, x_2) = 0.6 + \sum_{i=1}^n \left( \sin^2\left(1 - \frac{16}{15}x_i\right) - \frac{1}{50} \cdot \sin\left(4 - \frac{64}{15}x_i\right) - \sin\left(1 - \frac{16}{15}x_i\right) \right)$	$[-1, 1]^n$	$n = 2$	4	

**Table A2.** Multimodal test functions  $J_8 - J_{14}$  used in the experimental study.

Function	Search Domain	Dimensionality	Optima Number	Graph
<b>Rastrigin49</b>				
$J_8(x_1, x_2) = \sum_{i=1}^n x_i^2 - 18 \cdot \cos(2\pi x_i)$	$[-1, 1]^n$	$n = 2$	8	
<b>Roots</b>				
$J_9(x_1, x_2) = -\frac{1}{1 +  (x_1 + x_2)^6 - 1 }$	$[-2, 2]^n$	$n = 2$	6	
<b>Vincent</b>				
$J_{10}(x_1, x_2) = -\sum_{i=1}^n \sin(10 \cdot \log(x_i))$	$[0.25, 10]^n$	$n = 2$	36	
<b>Multi Peak</b>				
$J_{11}(x_1, x_2) = x_1 \cdot \sin(4\pi x_1) - x_2 \cdot \sin(4\pi x_2 + \pi) + 1$	$[-2, 2]^n$	$n = 2$	40	
<b>Alpine 02</b>				
$J_{12}(x_1, x_2) = \prod_{i=1}^n \sqrt{x_i} \sin(x_i)$	$[0, 10]^n$	$n = 2$	8	
<b>Cosine Mixture</b>				
$J_{13}(x_1, x_2) = -0.1 \sum_{i=1}^n \cos(5\pi x_i) - \sum_{i=1}^n x_i^2$	$[-1, 1]^n$	$n = 2$	12	
<b>Egg Crate</b>				
$J_{14}(x_1, x_2) = x_1^2 + x_2^2 + 25(\sin^2(x_1) + \sin^2(x_2))$	$[-5, 5]^n$	$n = 2$	9	

## References

1. Rao, S.S. *Engineering Optimization: Theory and Practice: Fourth Edition*; University of Miami: Florida, FL, USA, 2009.
2. Yang, X.-S.; Wiley InterScience (Online service). *Engineering Optimization: An Introduction with Metaheuristic Applications*; John Wiley & Sons: New Jersey, NJ, USA, 2010.
3. Persson, J.A.; Davidsson, P.; Johansson, S.J.; Wernstedt, F. *Combining Agent-Based Approaches and Classical Optimization Techniques*; Springer: Brussels, Belgium, 2005.
4. Cuevas, E.; Gálvez, J.; Hinojosa, S.; Avalos, O.; Zaldívar, D.; Pérez-Cisneros, M. A Comparison of Evolutionary Computation Techniques for IIR Model Identification. *J. Appl. Math.* **2014**, *2014*, 9. [[CrossRef](#)]
5. Lera, D.; Sergeyev, Y.D. Lipschitz and Hölder global optimization using space-filling curves. *Appl. Numer. Math.* **2010**, *60*, 115–129. [[CrossRef](#)]
6. Rana, P.B.; Patel, J.L.; Lalwani, D.I. Parametric optimization of turning process using evolutionary optimization techniques—A review (2000–2016). *Adv. Intell. Syst. Comput.* **2019**, *817*, 165–180.
7. Holland, J.H. *Adaptation in Natural and Artificial Systems*; University of Michigan Press: Michigan, MI, USA, 1975.
8. Goldberg, D.E. *Genetic Algorithms in Search, Optimization and Machine Learning*; Addison-Wesley: Reading, MA, USA, 1989.
9. Dorigo, M.; Stützle, T. The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances. In *Handbook of Metaheuristics*; Kluwer Academic Publishers: Boston, MA, USA, 2003; pp. 250–285.
10. Kennedy, J.; Eberhart, R.C. Particle swarm optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; pp. 1942–1948.
11. Karaboga, D. *An Idea Based on Honey Bee SWARM for Numerical Optimization*; Erciyes University: Kayseri, Turkey, 2005.
12. Rashedi, E.; Nezamabadi-pour, H.; Saryazdi, S. GSA: A Gravitational Search Algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [[CrossRef](#)]
13. Storn, R.; Price, K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [[CrossRef](#)]
14. Hansen, N.; Kern, S. Evaluating the CMA Evolution Strategy on Multimodal Test Functions. In Proceedings of the 8th International Conference Parallel Problem Solving from Nature—PPSN VIII, Birmingham, UK, 18–44 September 2004; Volume 3242/2004, pp. 282–291, no. 0.
15. Biswas, S.; Kundu, S.; Das, S. Inducing Niching Behavior in Differential Evolution Through Local Information Sharing. *IEEE Trans. Evol. Comput.* **2015**, *19*, 246–263. [[CrossRef](#)]
16. Sareni, B.; Krahenbuhl, L. Fitness sharing and niching methods revisited. *IEEE Trans. Evol. Comput.* **1998**, *2*, 97–106. [[CrossRef](#)]
17. Thomsen, R. Multimodal optimization using crowding-based differential evolution. In Proceedings of the Congress on Evolutionary Computation (CEC '04), Portland, OR, USA, 19–23 June 2004; pp. 1382–1389.
18. Liang, J.J.; Qu, B.Y.; Mao, X.B.; Niu, B.; Wang, D.Y. Differential evolution based on fitness Euclidean-distance ratio for multimodal optimization. *Neurocomputing* **2014**, *137*, 252–260. [[CrossRef](#)]
19. Biswas, S.; Das, S.; Kundu, S.; Patra, G.R. Utilizing time-linkage property in DOPs: An information sharing based Artificial Bee Colony algorithm for tracking multiple optima in uncertain environments. *Soft Comput.* **2014**, *18*, 1199–1212. [[CrossRef](#)]
20. Gao, W.; Yen, G.G.; Liu, S. A Cluster-Based Differential Evolution with Self-Adaptive Strategy for Multimodal Optimization. *IEEE Trans. Cybern.* **2014**, *44*, 1314–1327. [[CrossRef](#)]
21. Liang, Y.; Leung, K.-S. Genetic Algorithm with adaptive elitist-population strategies for multimodal function optimization. *Appl. Soft Comput.* **2011**, *11*, 2017–2034. [[CrossRef](#)]
22. Li, L.; Tang, K. History-Based Topological Speciation for Multimodal Optimization. *IEEE Trans. Evol. Comput.* **2015**, *19*, 136–150. [[CrossRef](#)]
23. Yao, J.; Kharna, N.; Zhu, Y.Q. On Clustering in Evolutionary Computation. In Proceedings of the IEEE International Conference on Evolutionary Computation, Vancouver, BC, Canada, 16–21 July 2006; pp. 1752–1759.
24. Ursem, R.K. Multinational evolutionary algorithms. In Proceedings of the Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Washington, DC, USA, 6–9 July 1999; pp. 1633–1640.

25. Yao, J.; Kharmar, N.; Grogono, P. Bi-Objective Multipopulation Genetic Algorithm for Multimodal Function Optimization. *IEEE Trans. Evol. Comput.* **2010**, *14*, 80–102.
26. Deb, K.; Saha, A. Multimodal Optimization Using a Bi-Objective Evolutionary Algorithm. *Evol. Comput.* **2012**, *20*, 27–62. [[CrossRef](#)] [[PubMed](#)]
27. Basak, A.; Das, S.; Tan, K.C. Multimodal Optimization Using a Biobjective Differential Evolution Algorithm Enhanced with Mean Distance-Based Selection. *IEEE Trans. Evol. Comput.* **2013**, *17*, 666–685. [[CrossRef](#)]
28. Wang, Y.; Li, H.X.; Yen, G.G.; Song, W. MOMMOP: Multiobjective Optimization for Locating Multiple Optimal Solutions of Multimodal Optimization Problems. *IEEE Trans. Cybern.* **2015**, *45*, 830–843. [[CrossRef](#)]
29. Gálvez, J.; Cuevas, E.; Becerra, H.; Avalos, O. A hybrid optimization approach based on clustering and chaotic sequences. *Int. J. Mach. Learn. Cybern.* **2020**, *11*, 359–401. [[CrossRef](#)]
30. Murtagh, F.; Legendre, P. Ward's Hierarchical Agglomerative Clustering Method: Which Algorithms Implement Ward's Criterion? *J. Classif.* **2014**, *31*, 274–295. [[CrossRef](#)]
31. Couzin, I.D.; Krause, J.; James, R.; Ruxton, G.D.; Franks, N.R. Collective memory and spatial sorting in animal groups. *J. Theor. Biol.* **2002**, *218*, 1–11. [[CrossRef](#)]
32. Ballerini, M.; Cabibbo, N.; Candelier, R.; Cavagna, A.; Cisbani, E.; Giardina, I.; Lecomte, V.; Orlandi, A.; Parisi, G.; Procaccini, A.; et al. Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study. *Proc. Natl. Acad. Sci. USA* **2008**, *105*, 1232–1237. [[CrossRef](#)]
33. Cuevas, E.; González, M.; Zaldivar, D.; Pérez-Cisneros, M.; García, G. An Algorithm for Global Optimization Inspired by Collective Animal Behavior. *Discret. Dyn. Nat. Soc.* **2012**, *2012*, 1–24. [[CrossRef](#)]
34. Caponetto, R.; Fortuna, L.; Fazzino, S.; Xibilia, M.G. Chaotic Sequences to Improve the Performance of Evolutionary Algorithms. *IEEE Trans. Evol. Comput.* **2003**, *7*, 289–304. [[CrossRef](#)]
35. Zelinka, I.; Lampinen, J.; Senkerik, R.; Pluhacek, M. Investigation on evolutionary algorithms powered by nonrandom processes. *Soft Comput.* **2018**, *22*, 1791–1801. [[CrossRef](#)]
36. Mozaffari, A.; Emami, M.; Fathi, A. A comprehensive investigation into the performance, robustness, scalability and convergence of chaos-enhanced evolutionary algorithms with boundary constraints. *Artif. Intell. Rev.* **2019**, *52*, 2319–2380. [[CrossRef](#)]
37. He, D.; He, C.; Jiang, L.G.; Zhu, H.W.; Hu, G.R. Chaotic characteristics of a one-dimensional iterative map with infinite collapses. *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.* **2001**, *48*, 900–906.
38. Lawnik, M. Generation of Numbers with the Distribution Close to Uniform with the Use of Chaotic Maps. In Proceedings of the 4th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH), Vienna, Austria, 28–30 August 2014; pp. 451–455.
39. De Jong, K.A. An Analysis of the Behavior of a Class of Genetic Adaptive Systems. Ph.D. Thesis, University of Michigan, Michigan, MI, USA, 1975.
40. Gálvez, J.; Cuevas, E.; Avalos, O. Flower Pollination Algorithm for Multimodal Optimization. *Int. J. Comput. Intell. Syst.* **2017**, *10*, 627. [[CrossRef](#)]
41. Li, X.; Engelbrecht, A.; Epitropakis, M. *Benchmark Functions for CEC 2013 Special Session and Competition on Niching Methods for Multimodal Function Optimization*; R. Melb. Inst. ... ; RMIT University: Melbourne, Australia, 2013; pp. 1–10.
42. Chitsaz, H.; Amjady, N.; Zareipour, H. Wind power forecast using wavelet neural network trained by improved Clonal selection algorithm. *Energy Convers. Manag.* **2015**, *89*, 588–598. [[CrossRef](#)]
43. Aung, T.N.; Khaing, S.S. *Genetic and Evolutionary Computing*; Springer: New York, NY, USA, 2016.
44. Qu, B.Y.; Suganthan, P.N.; Das, S. A distance-based locally informed particle swarm model for multimodal optimization. *IEEE Trans. Evol. Comput.* **2013**, *17*, 387–402. [[CrossRef](#)]
45. De Castro, L.; Von Zuben, F. The clonal selection algorithm with engineering applications. In Proceedings of the GECCO, Las Vegas, NV, USA, 8–12 July 2000; pp. 36–37.
46. Vollmer, D.T.; Soule, T.; Manic, M. A distance measure comparison to improve crowding in multi-modal optimization problems. In Proceedings of the ISRCS 2010—3rd International Symposium on Resilient Control Systems, Idaho Falls, ID, USA, 10–12 August 2010; pp. 31–36.
47. Mahfoud, S.W. Niching Methods for Genetic Algorithms. Ph.D. Thesis, University of Illinois at Urbana-Champaign, Urbana Champaign, IL, USA, 1995.
48. Biswas, S.; Kundu, S.; Das, S. An improved parent-centric mutation with normalized neighborhoods for inducing niching behavior in differential evolution. *IEEE Trans. Cybern.* **2014**, *44*, 1726–1737. [[CrossRef](#)]

49. Yazdani, S.; Nezamabadi-pour, H.; Kamyab, S. A gravitational search algorithm for multimodal optimization. *Swarm Evol. Comput.* **2014**, *14*, 1–14. [[CrossRef](#)]
50. Hui, S.; Suganthan, P.N. Ensemble and Arithmetic Recombination-Based Speciation Differential Evolution for Multimodal Optimization. *IEEE Trans. Cybern.* **2016**, *46*, 64–74. [[CrossRef](#)] [[PubMed](#)]
51. Lacroix, B.; Molina, D.; Herrera, F. Region-based memetic algorithm with archive for multimodal optimization. *Inf. Sci.* **2016**, *367*, 719–746. [[CrossRef](#)]
52. Wilcoxon, F. Individual comparisons by ranking methods. *Biometrics* **1945**, *1*, 80–83. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).