

Article

A Novel Evolutionary Algorithm: One-Dimensional Subspaces Optimization Algorithm (1D-SOA)

Gabriela Berenice Díaz-Cortés ¹ and René Luna-García ^{2,*}

¹ Instituto Mexicano del Petróleo, Eje Central Lázaro Cárdenas 152, Col. San Bartolo Atepehuacan, Mexico City 07730, Mexico; gbdiaz@imp.mx

² Centro de Investigación en Computación, Instituto Politécnico Nacional, Esq. Miguel Othón de Mendizábal, Av. Juan de Dios Bátiz, Col. Nueva Industrial, Mexico City 07738, Mexico

* Correspondence: rlunag@ipn.mx

Abstract: This paper introduces an evolutionary algorithm for n-dimensional single objective optimization problems: One-Dimensional Subspaces Optimization Algorithm (1D-SOA). The algorithm starts with an initial population in randomly selected positions. For each individual, a percentage of the total number of dimensions is selected, each dimension corresponding to a one-dimensional subspace. Later, it performs a symmetric search for the nearest local optima in all the selected one-dimensional subspaces (1D-S), for each individual at a time. The search stops if the new position does not improve the value of the objective function over all the selected 1D-S. The performance of the algorithm was compared against 11 algorithms and tested with 30 benchmark functions in 2 dimensions (D) and 30D. The proposed algorithm showed a better performance than all other studied algorithms for large dimensions.

Keywords: single-objective optimization; evolutive algorithms; metaheuristics



Citation: Díaz-Cortés, G.B.; Luna-García, R. A Novel Evolutionary Algorithm: One-Dimensional Subspaces Optimization Algorithm (1D-SOA). *Symmetry* **2023**, *15*, 1873. <https://doi.org/10.3390/sym15101873>

Academic Editors: Peng-Yeng Yin, Ray-I Chang and Jen-Chun Lee

Received: 9 June 2023

Revised: 3 October 2023

Accepted: 3 October 2023

Published: 5 October 2023



Copyright: © 2023 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Everyday life requires finding optimal solutions in a faster way to reduce costs and computational times, and to increase productivity or revenues. Finding the best or less costly route, setting up an optimal schedule, reducing production times, and controlling engineering systems are some examples of optimization problems. Therefore, optimization techniques are relevant in broad areas of life, which makes the continuous research and improvement in existing algorithms necessary [1–4].

Global optimization is concerned with finding the global solution for which the objective function obtains its smallest value. Formally, global optimization seeks a global solution of a constrained optimization model.

When only one function is optimized, the problem is referred to as a single objective optimization problem, while if two or more fitness functions are optimized, it is called a multiobjective problem.

The fitness function is regarded as a mapping $\vec{x} \rightarrow f(\vec{x})$, where $\vec{x} \in \mathbb{R}^n$ is a multi-dimensional decision variable, and $f(\vec{x}) \in \mathbb{R}$ is part of the objective space Ω . Thus, we have a mapping $\mathbb{R}^n \rightarrow \mathbb{R}$. Given a function $f(\vec{x}) \in \mathbb{R}$, the solution to the optimization problem is a vector \vec{x}^* , such that the fitness function is close to its optimum value, i.e., it satisfies $|f(\vec{x}^*) - f(\vec{x}_{min/max})| < \epsilon$, where $\vec{x}_{min/max}$ is the vector that gives the minimum or maximum value of the objective function, that is, the vector that minimizes or maximizes the objective function.

Among others, metaheuristic methods are efficient techniques to solve optimization problems [5]. These methods use an iterative process to improve an initial solution up to a selected termination criterion, within a reasonable time and with low computational costs. There are several classifications for metaheuristics; Beheshti et al. [6] classify them as nature [1,7–17] and non-nature inspired [2,18–20], population-based, single point

search, dynamic and static objective function, single and various neighborhood structures, and memory usage and memory-less methods, among others.

Population-based metaheuristic algorithms have been broadly studied for global searches, as they can handle high-dimensional optimization problems. They present exploration and exploitation capabilities where exploration is related to the generation of new individuals in unexplored regions of the search space, while exploitation focuses the search in the neighborhood of known solutions [21]. Too much of the former can lead to an inefficient search, and too much of the latter can lead to a propensity to focus the search too quickly, losing many possible solutions [22].

A good balance between exploration and exploitation is required to avoid the trapping of the particles into local optima (premature convergence), and to find a good optimal solution in a few iterations [23]. Recent works try to incorporate local search with exploration to achieve the better performance of the algorithms. The most popular heuristic algorithms are the particle swarm optimization algorithm (PSO) [18,24,25] and the Differential Evolution Algorithm [26,27], which have been modified numerous times to improve their local and global search capabilities [28–30]. In a study by Sun et al. [31], the Whale Optimization Algorithm (WOA) is modified to perform a more detailed local search. The Butterfly Optimization Algorithm (BOA) is also improved by Li et al. [1] to achieve a balance between exploration and exploitation. The Polar Bear Optimization Algorithm (PBO) [9] was proposed, taking into account an efficient birth and death mechanism to improve global and local search.

In recent years, new algorithms have been proposed with new local and global search models, some examples of these are: Artificial ecosystem-based optimization (AEO) [32], Jellyfish-inspired metaheuristic (JF) [33], Chaos Game Optimizer (CGO) [34], Zebra Optimization Algorithm (ZOA), [35], Chameleon Swarm Algorithm (CSA) [36], Serval Optimization Algorithm (SOA) [37], optimization problems based on walrus behavior (OWB) [38], LSHADE-SPACMA Algorithm [39], Coronavirus Optimization Algorithm (COA) [40], Gaining-Sharing Knowledge Based Algorithm With Adaptive Parameters (APKBS) [41], and Improving Multi-Objective Differential Evolution Algorithm (IMODE) [42], etc.

Some of the most efficient metaheuristics are the Evolutionary Algorithms (EAs), which are flexible enough to solve different types of problems due to their exploration and exploitation capabilities. They are distinguished from Genetic Algorithms (GA) in the way they combine information through evolutionary operators that evolve the population, obtaining a set of new solutions [43]. These methods can solve problems of high computational complexity or high dimensional problems.

This work proposes a novel optimization method of the evolutionary type: the One-Dimensional Subspaces Optimization Algorithm (1D-SOA). This algorithm performs an exhaustive local search or exploitation, by updating the position of each individual of a randomly generated population, over a selected direction. This search generates a large number of optimal local sets. The search neighborhood is changed randomly, increasing, in this way, the local search space. Nevertheless, this is not the only characteristic of the algorithm. During the whole process, it also performs exploration, by generating diversity via the recombination of the initial population, which favors a convergence towards the global optimum. The algorithm is tested for single-objective optimization problems. More details of the algorithm are presented in Section 3.

This work is structured as follows, in Section 2, the theory related to single objective optimization problems is presented. Later, in Section 3, 1D-SOA is introduced, together with details about its implementation and some EAs used for comparison. After that, the results and discussion are presented in Section 4. Finally, the conclusions and references are given in Section 5.

2. Optimization

In this section, the theory of unconstrained optimization is presented. Additionally, some concepts related to metaheuristic optimization are introduced.

2.1. Unconstrained Single Objective Optimization

Unconstrained single-objective optimization problems are defined in terms of the decision variables, the objective function to be optimized, and the set of feasible solutions. The set of all feasible solutions is called the feasible region or search space, and it is commonly denoted by Ω . The decision variables are numerical quantities related to the optimization problem. These quantities are denoted as x_d with $d = 1, 2, \dots, D$ [44,45]. The vector $\vec{x} \in \Omega$ composed of D decision variables is represented by the following:

$$\vec{x} = [x_1, x_2, \dots, x_D]^T. \quad (1)$$

The objective function, which is a function of the decision variables, is minimized or maximized according to the problem's criteria. The restrictions are related to the limits of the variables, that is, they restrict each decision variable x_d in the decision space, within a lower range $x_d^{(L)}$ and an upper range $x_d^{(U)}$. The formal definition is presented below.

The unconstrained single-objective optimization problem is defined as follows:

$$\begin{aligned} & \text{Min/Max } f(\vec{x}), \\ & x_d^{(L)} \leq x_d \leq x_d^{(U)}. \end{aligned} \quad (2)$$

If the objective function is linear, it is considered a linear problem, otherwise, it is non-linear [46]. This work addresses unconstrained non-linear optimization problems; in particular, the optimization of a set of benchmark functions is studied.

2.2. Metaheuristics

Metaheuristics are optimization methods that combine local improvement procedures and high-level strategies to escape from local optima, performing a robust search in the solution space. These algorithms have stochastic behavior and mimic biological or physical processes. The fundamental phases of the metaheuristics are as follows [47]:

1. Initialize population;
2. Define stop condition;
3. Evaluate fitness function;
4. Update and move agents;
5. Return the global best solution.

Each metaheuristic defines a particular movement of the population according to the process they want to mimic. Depending on the part of the search space that the method covers, the algorithm can find local or global optima. Next, some concepts related to the search methodology performed by some metaheuristics, in particular used by 1D-SOA, are presented.

2.2.1. Local Search

Local search algorithms explore the neighborhood of the current point to find a better solution, i.e., the local optimum. The search starts with a random feasible point, then, a generation or movement mechanism is successively applied to optimize the fitness function. If a better solution is found, it becomes the current solution. This procedure is repeated up to a stopping criterion. Local search heuristics select one or, at most, two neighborhoods for improving the current solution (i.e., $k_{max} \leq 2$), and they converge to local optima [3].

2.2.2. Exploration

EAs usually converge to local optima. However, the search space may contain much better solutions in other regions. Additionally, the optima can be duplicated, leading to uniformity. Therefore, to find an optimal solution, the EAs should keep moving to unexplored regions trying to reach a better solution. This can be done by mutating duplicates,

via the use of a recombination operation to replace part of the population [23]. The process of moving to distant or unexplored regions is commonly known as exploration.

2.2.3. Variable Neighborhood Search

Variable neighborhood search (VNS) is a metaheuristic for solving global optimization problems, based on a systematic change of neighborhood. This method continues the search after finding the first local minimum by selecting a new neighborhood to escape the local valley.

The neighborhoods search is based on the empirical idea that a local optimum can be related to the global one, for example by sharing some components of the position. Therefore, a study of the neighborhoods of a local optimum can lead to the discovery of a better one. The neighborhoods are denoted as N_k , $k = 1, \dots, k_{max}$, where the selection of a neighborhood depends on the framework, which can be stochastic, deterministic, or a combination of both [3]. Reaching the global optimum is more likely if the search is performed in various neighborhoods rather than a single structure.

2.2.4. Swarm Intelligence

Swarm Intelligence (SI) is a branch of Computational Intelligence (CI) that originated from the study of the social behavior of insects, animals, and human societies (individuals) [3,47,48]. A SI system consists of a population of individuals or agents that controls their behavior autonomously, and each individual represents a possible solution.

For the swarm, there is no central coordination and it can consist of a few to up to thousands of agents interacting among themselves and with the environment, without a change of control architecture. Therefore, SI systems are robust and scalable. Furthermore, no single agent is essential for the swarm, which gives the swarm its characteristic flexibility [3,23,47]. SI algorithms evolve a population of possible problem solutions which improve with each iteration, therefore, they are considered to be an EA [23]. The method introduced in this work is a population-based evolutionary algorithm that performs a search for better solutions on a randomly selected dimension d . The particles present communication among themselves, which makes it a Swarm Intelligence system, details about the algorithm are presented in the next section.

3. Methodology

In this section, the basic ideas behind 1D-SOA, together with the implementation of the algorithm, are presented.

1D-SOA:

1. *Initialization*: 1D-SOA starts by randomly generating an initial D -dimensional population within the given search space. The fitness value of each individual or particle $\vec{x}_i \in \mathbb{R}^D$ is computed, and the best particle is spotted \vec{x}^* . The initial population is a set of N individuals, denoted by $\mathbf{P}(0) = \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N\}$, with

$$\vec{x}_i = [x_{i,1}, x_{i,2}, \dots, x_{i,d}, \dots, x_{i,D}]^T \in \Omega, \quad (3)$$

where the subscript i, d , where $d \in [1, D]$ in $x_{i,d}$ represents the d -th component of the i -th particle \vec{x}_i .

2. *Symmetric selection of search direction d_{off}* : Once the initial population is constructed, a dimension d_{off} is chosen, and the position of the particles is updated a step Δ over the selected 1D-S. To select the dimension, D_{test} directions are randomly generated and tested by moving the particle *steps* times symmetrically. The direction with the best value of the objective function is selected as the search direction (d_{off}). The values D_{test} and *steps* can be selected by the user.
3. *Step Δ* : 1D-SOA includes a Variable Neighborhood Search (VNS) with two neighborhoods (N_1 and N_2). The size of N_1 depends on a parameter C_{neigh} . The step in this

neighborhood is chosen randomly according to Gaussian distribution with mean 0 and variance C_{neigh} .

$$\Delta = |rand[0, C_{neigh}]| \in N_1.$$

For N_2 , a smaller random parameter C_{neigh_1} , is used with mean 0 and variance C_{neigh_1} .

$$\Delta = |rand[0, C_{neigh_1}]| \in N_2.$$

This second neighborhood performs a more local search. The neighborhood is selected for each particle, depending on a random criterion denoted as Pns , referred to as the probability of neighborhood search. For the selection, a random number from a uniform distribution from 0 to 1 is obtained; if the number is smaller than Pns , the search is performed in N_1 , otherwise it is performed in N_2 . This criterion is presented in Equation (4):

$$\Delta = \begin{cases} \Delta \in N_1, & \text{if } rand[0, 1] < Pns, \\ \Delta \in N_2, & \text{otherwise.} \end{cases} \quad (4)$$

Once the neighborhood is selected, the value of the delta is updated as follows:

$$\Delta = \Delta * |rand[0, 1]|.$$

4. *Asymmetric movement operator*: The position of the particle varies asymmetrically during the iteration process. That is, it moves only in the direction where the objective function improves. For a particle \vec{x}_i at the t -th iteration, it is denoted by

$$\vec{x}_i(t) = x_{i,1}(t)\hat{e}_1 + x_{i,2}(t)\hat{e}_2 + \cdots + x_{i,d}(t)\hat{e}_d + \cdots + x_{i,D}(t)\hat{e}_D, \quad (5)$$

where \hat{e}_d is the canonical normal vector in the d -th dimension. Similarly, the position of the same particle at time $t + 1$ is

$$\vec{x}_i(t+1) = x_{i,1}(t)\hat{e}_1 + x_{i,2}(t)\hat{e}_2 + \cdots + x_{i,d}(t+1)\hat{e}_d + \cdots + x_{i,D}(t)\hat{e}_D. \quad (6)$$

The position of each individual \vec{x}_i is updated over the selected dimension (see Equation (6)). That is, for a dimension d_{off} , with $1 < d_{off} < D$, there is a change on the current d_{off} component of the position vector of the individuals $x_{i,d_{off}}(t) \rightarrow x_{i,d_{off}}(t+1)$ as follows:

$$x_{i,d_{off}}(t+1) = x_{i,d_{off}}(t) \pm \Delta. \quad (7)$$

The sign (+) or (−) of the update corresponds to the direction where the objective function improves. Combining Equations (6) and (7), the updated position at the $(t + 1)$ -th iteration is given by

$$\vec{x}(t+1) = x_1(t)\hat{e}_1 + x_2(t)\hat{e}_2 + \cdots + (x_d(t) \pm \Delta)\hat{e}_d + \cdots + x_D(t)\hat{e}_D. \quad (8)$$

If the particle does not improve its objective function after two consecutive movements by a value ϵ , that is

$$(f(\vec{x}(t+1)) - f(\vec{x}(t))) < \epsilon, \quad (9)$$

the value of the delta is updated as

$$\Delta = \Delta/2. \quad (10)$$

5. *Semi cross*: The probability of semi-cross is denoted by Psx ; if this criterion is satisfied, the worst $Nparx$ particles are selected. Later, the d_{off} -th component of the i -th particle,

$x_{i,d_{off}}$, $i \in \{1, \dots, N_{parx}\}$, is substituted by the d_{off} -th component of the best particle $x_{d_{off}}^*$, as presented in Equation (11)

$$x_{i,d_{off}} = \begin{cases} x_{d_{off}}^* & \text{if } rand[0,1] < P_{sx} \\ x_{i,d_{off}} & \text{otherwise} \end{cases}, \quad i \in \{1, \dots, N_{parx}\}. \quad (11)$$

6. **Recombination:** The worst $N_{rec} = Nr * D$ particles of the new population $P(i+1)$, where Nr is the percentage of the initial population to recombine, is replaced by a set of recombined elements if the new elements lead to a better solution. The new population becomes

$$P(i+1) = \{x_1(i+1), x_2(i+1), \dots, x_r(i+1), \dots, x_t(i+1)\},$$

where $x_r(i+1), \dots, x_t(i+1)$ are recombined elements.

Recombination operator: Each selected element $x_\alpha \in P$ is recombined as follows,

$$R(\vec{x}_\alpha) = \vec{x}_\alpha - r_1(\vec{x}^* - \vec{x}_\alpha),$$

where r_1 is a random number between 0 and 1.

7. **Stopping criterion:** The stopping criterion is a selected number of generations.
8. **Solutions:** After the optimization cycle, a set **S** of optimal solutions of the problem is obtained

$$S = \{x_1^{sol}, x_2^{sol}, \dots, x_{N-1}^{sol}, x_N^{sol}\}.$$

The implementation of 1D-SOA is presented in Algorithm 1 and in Figure 1. The definition of the variables presented in this section and in Algorithm 1 are presented in Table 1. It is worth mentioning that, for the variables required for the 1D-SOA algorithm, only two of them change with the problem, the rest can be taken as fixed in any optimization problem.

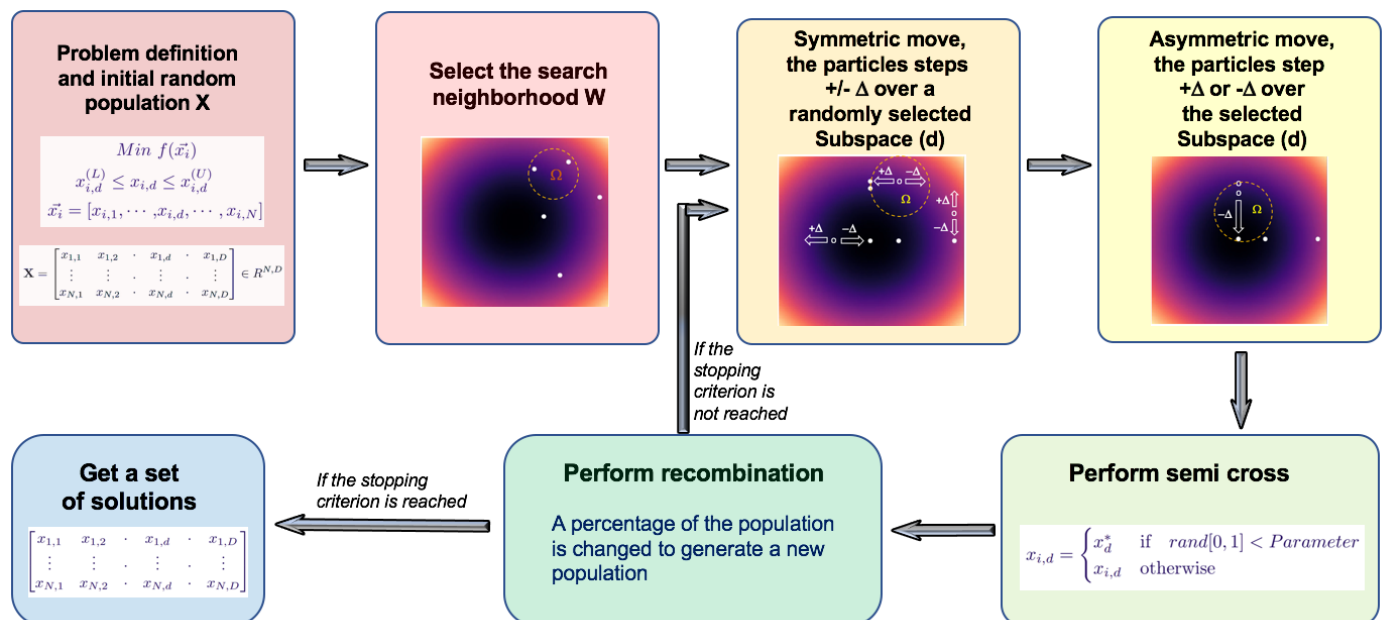


Figure 1. Basic algorithmic structure of 1D-SOA.

Algorithm 1 One-Dimensional Search Optimization Algorithm (1D-SOA)**Input:** $N, D, Max_G, Pns, N_{dims}, Max_\epsilon, Psx, Nparx, Nr$ for the definitions see Table 1.**Output:** A set of optimal solutions S .

```

1:  $P(0) \rightarrow P$  % Create a random initial population  $P(0)$  of  $N$  individuals ( $\vec{x}_i \in \mathbb{R}^D$ ).
2: for generation = 1 to  $G$  do
3:   for particle = 1 to  $N$  do
4:     if  $UniformRand[0,1] < Pns$  then
5:        $\Delta \in N_1$ 
6:     else
7:        $\Delta \in N_2$  % Select the search neighborhood
8:     end if
9:     for  $d = 1$  to  $N_{dims}$  do
10:      Symmetric selection of the search dimension  $d_{off}$ 
11:       $\Delta = \Delta * abs(GaussRand[0,1])$ . % Update the step  $\Delta$ 
12:      Asymmetric movement of the particles on dimension  $d_{off}$  in direction  $\pm \Delta$ .
13:      while ( $f(\vec{x}_i(t+1)) - f(\vec{x}_i(t)) < \epsilon \mid It_{epsilon} < Max_\epsilon$ ) do
14:         $x_{i,d_{off}}(t+1) = x_{i,d_{off}}(t) \pm \Delta$ 
15:      end while
16:      if  $UniformRand[0,1] < Psx$  then
17:        for particle in  $Nparx$  do
18:           $x_{i,d_{off}}(t+1) = x_{i,d_{off}}^*(t+1)$  % Perform semi cross
19:        end for
20:      end if
21:      for particle in  $Nr * D$  do
22:         $\vec{x}_\alpha = \vec{x}_\alpha - r_1(\vec{x}^* - \vec{x}_\alpha)$  % Perform recombination
23:      end for
24:    end for
25:  end for
26:  % Save the set  $S$  of solutions.
27: end for

```

Table 1. Definitions of variables.

Variable	Meaning	Value
Variables required for all the algorithms		
D	Dimension	2, 30, 60
N	Number of individuals	10 for 2, 5, and 10, D for the rest
Max_G	Maximum number of generations	100
$x_d^{(L)}$	Lower bound of \vec{x} in the dimension d	Depends on the function (Tables A8–A10)
$x_d^{(U)}$	Upper bound of \vec{x} in the dimension d	Depends on the function (Tables A8–A10)
$\vec{x}_{min} \in \mathbb{R}^N$	Vector that minimizes the objective function, optimal solution for a minimization problem	Depends on the function (Tables A8–A10)
$\vec{x}_i \in \mathbb{R}^N$	Particle	
Variables of 1D-SOA that depend on the problem		
C_{neigh}	Neighborhood parameter	Depends on the function (Tables A8–A10)
C_{neigh_l}	Local neighborhood parameter	Depends on the function (Tables A8–A10)
Variables of 1D-SOA that do not change		
D_{test}	Random test directions to select the direction with the largest improvement	20% D
$steps$	Number of steps given in each test direction	5
Max_ϵ	Maximum allowed difference between previous and actual point	$1e^{-2}$
Max_{NI}	Maximum allowed number of not improvements of the solution	20
Npx	Percentage of particles to perform semi cross on	50%
Psx	Probability of semi cross	20%
Pns	Probability of neighborhood search	10%
P_{dim}	Number of search dimensions	50% D
Nr	Percentage of particles to recombine	20% D

4. Results and Discussion

This section contains the results of the experiments performed to study the behavior of the 1D-SOA algorithm introduced in this work. The performance of 1D-SOA was compared against the heuristic algorithms from the python library SwarmPackagePy [49]; in particular, the studied algorithms from this library are Artificial Bee Algorithm (ABA) [50], Bat Algorithm (BA) [51], Bacterial Foraging Optimization (BFO) [52], Cat Swarm Optimization (CA) [17], Chicken Swarm Optimization (CHSO) [10], Cuckoo Search Optimization (CU) [13], Firefly algorithm (FA) [15], Fireworks Algorithm (FWA) [20], Gravitational Search Algorithm (GSA) [19], Grey Wolf Optimizer (GWO) [19], Particle Swarm Optimization (PSO) [18], and Social Spider Algorithm (SSA) [14]. Additionally, we include in the comparison some state-of-the-art algorithms: Mean particle Swarm Optimisation (MPSO) [53], Artificial ecosystem-based optimization (AEO) [32], Jellyfish inspired metaheuristic (JF) [33], Chaos Game Optimizer (CGO) [34], and Zebra Optimization Algorithm (ZOA) [35].

All algorithms require a basic number of input parameters: the number of individuals (N), dimension (D), number of maximum iterations (N_{Its}), and lower ($x^{(L)}$) and upper ($x^{(U)}$) limits in the search space. Additionally, some algorithms need a specific set of extra control parameters. Table 2 shows the specific parameters, together with the most common values used in the literature and this work. A brief discussion of the selection of parameters used for the MPSO, FWA, and CHSO is presented in Appendix A. For the 1D-SOA Algorithm, besides the parameters presented in the above-mentioned table, it is required to define the following fixed variables, $Npx = 0.5$, $Psx = 0.2$, $Pns = 0.1$, $Pdim = 0.5$, and $Nr = 0.2$ that we recommend not to change, as they work for all the studied problems. The 30 studied benchmark functions are presented in the Appendix C, Tables A8–A10. This table contains the search space, the optimal solution vector $\vec{x}^* = [x_1^*, x_2^*, \dots, x_D^*]$, and the function evaluated at this vector $f(\vec{x}^*)$, i.e., the optimum. Additionally, for the 1D-SOA method, it is required to select two parameters, C_{neigh} and C_{neigh1} ; the values of these parameters are presented in the same tables.

Table 2. Evolutionary algorithms used in this work, together with the extra parameters required.

Algorithm	Parameters
Bat Algorithm (BA) [51]	$r_0 = 0.9$, $V_0 = 0.5$, $f_{min} = 0$, $f_{max} = 0.02$, $alpha = 0.9$, $csi = 0.9$
Bacterial Foraging Optimization (BFO) [52]	$N_c = 2$, $N_s = 12$, $C = 0.2$, $Ped = 1.15$
Cat Swarm Optimization (CA) [17]	$Mr = 10$, $smp = 2$, $spc = False$, $cdc = 1$, $srd = 0.1$, $w = 0.1$, $c = 1.05$, $csi = 0.6$
Chicken Swarm Optimization (CHSO) [10]	$G = 5$, $FL = 0.5$
Cuckoo Search Optimization (CU) [13]	$pa = 0.25$, $nest = 100$
Firefly algorithm (FA) [15]	$csi = 1$, $psi = 1$, $alpha_0 = 1$, $alpha_1 = 0.1$, $norm_0 = 0$, $norm_1 = 0.1$
Fireworks Algorithm (FWA) [20]	$m_1 = 7$, $m_2 = 7$, $eps = 0.001$, $amp = 2$, $a = 0.3$, $b = 3$
Gravitational Search Algorithm (GSA) [19]	$G_0 = 3$
Mean particle Swarm Optimisation (MPSO) [53]	$w = 0.7$, $c_1 = 1$, $c_2 = 1$.
Particle Swarm Optimization (PSO) [18]	$w = 0.7$, $c_1 = 1$, $c_2 = 1$.
Social Spider Algorithm (SSA) [14]	$pf = 0.4$
One-Dimensional Search Algorithm (1D-SOA)	C_{neigh} , C_{neigh1} , see Tables A8–A10

The comparison was made for different dimensions, in particular $D = \{2, 30\}$.

For the experiments, a statistical analysis of the performance of the algorithms was performed and the algorithms were compared from 100 runs for each function. From the 100 runs, the best solution and the mean of the 100 runs were obtained for the statistic at an approximated value of $1e^{-8}$; for the functions that do not reach this value, the 100 iterations were taken.

The initial population varied for each dimension; for dimensions less or equal to 10 the initial population was 10, while for larger dimensions the initial population corresponds to the dimension. Next, we present the results for dimensions 2 and 30.

In Figure 2, all the algorithms are presented for 30 D for the ackley, ackley_2, rastrigin, and schwefel 2.20 functions. It can be noticed that the algorithms with slightly better performance are: 1D-SOA, FWA, ZOA, AEO, MPSO, and CHSO. For the rest of the algorithms, most of the time they do not converge in the required number of iterations. Therefore, to have more clarity in the plots and tables, only the algorithms mentioned above are presented. In Figures 3 and 4, the convergence for different steps for six search directions, and for various search directions and five steps, are presented. It is worth noticing from the plots that the convergence slightly improves with more search steps and more search directions. However, the computational work increases, therefore the parameters selected were five steps and 20% of the dimensions as search dimensions, as these parameters have a good balance between work and performance (see Table 1).

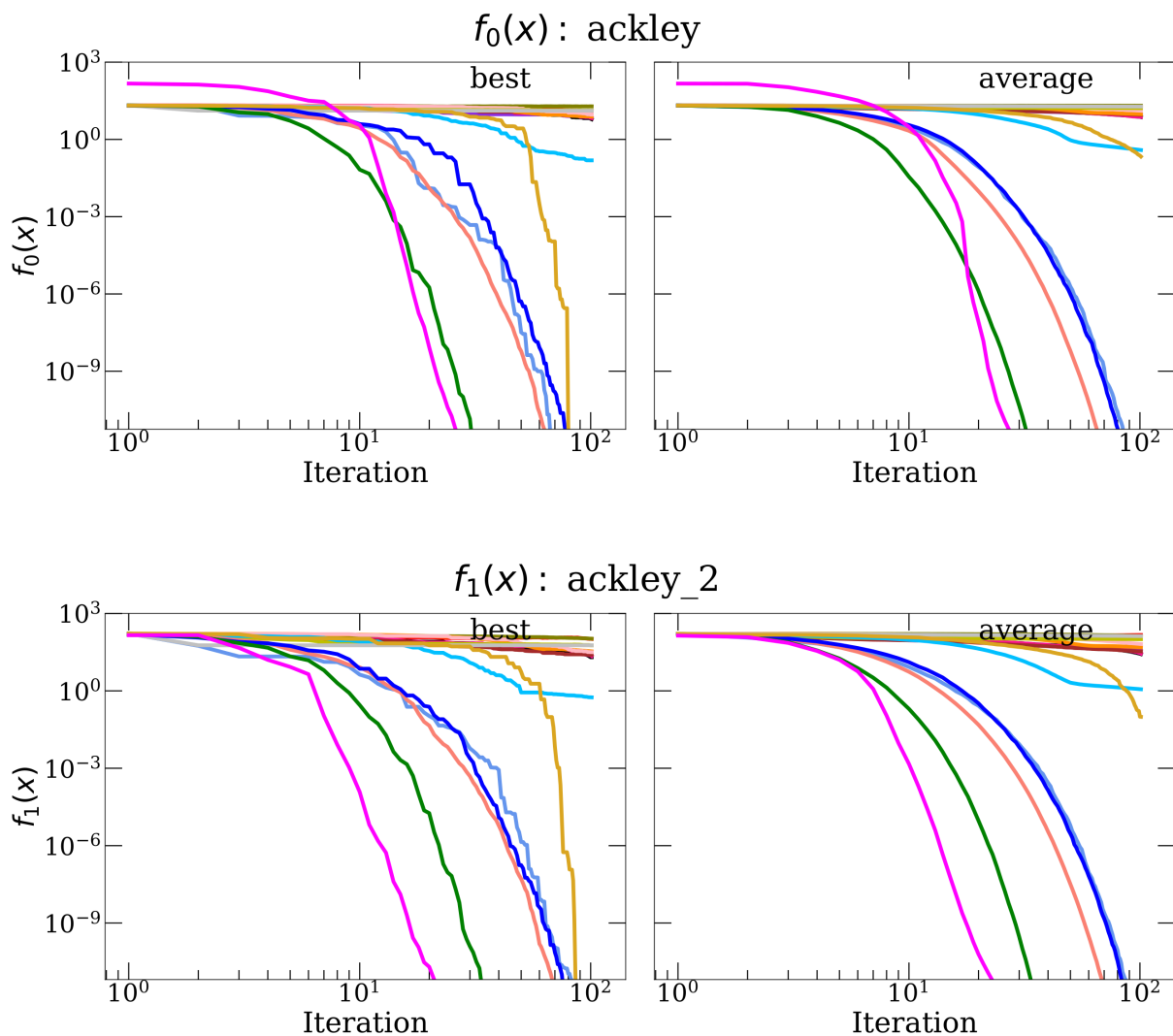


Figure 2. Cont.

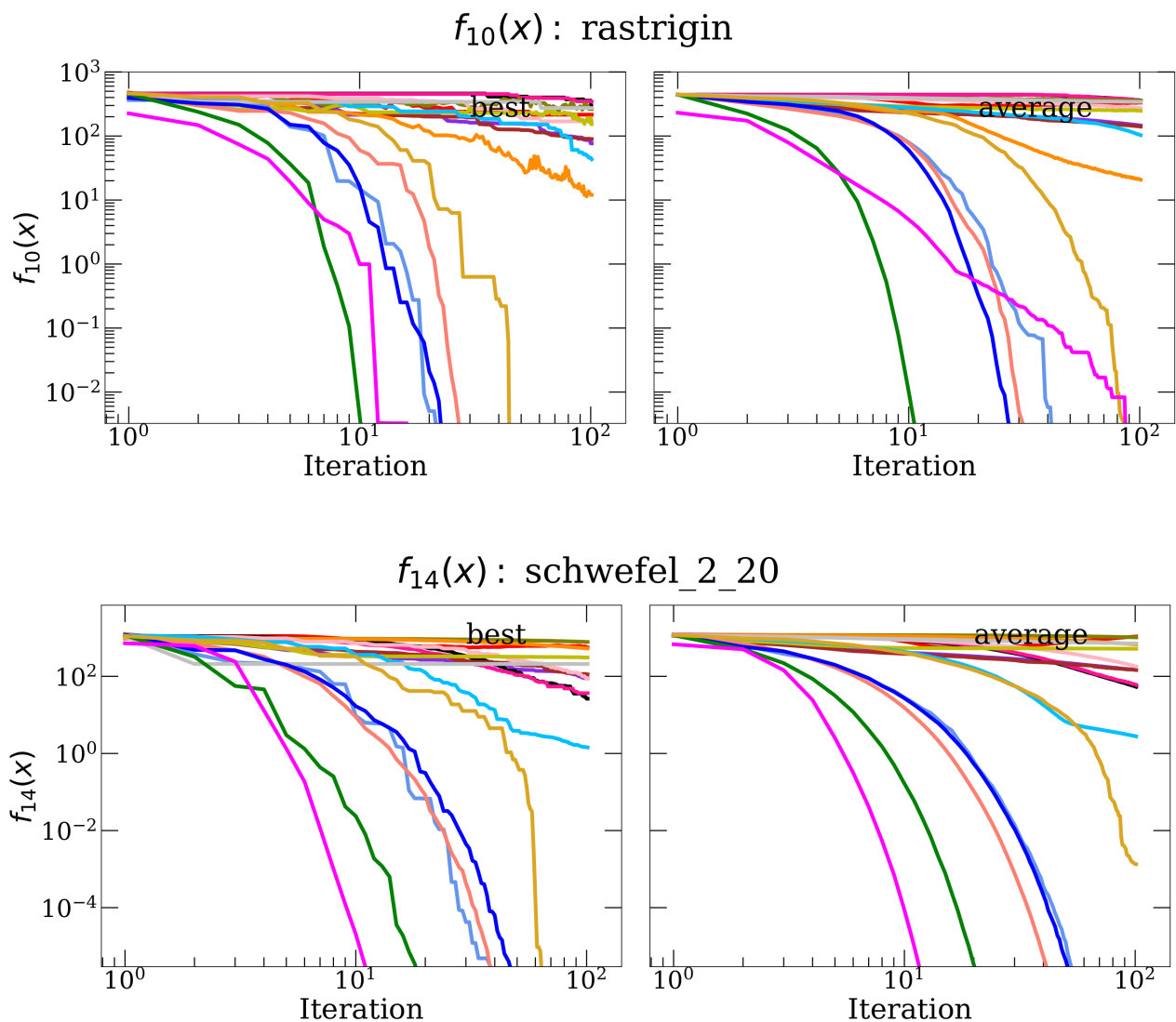


Figure 2. Convergence of the ackley, ackley_2, rastrigin, and schwefel 2.20 functions for all the algorithms in 30 D.

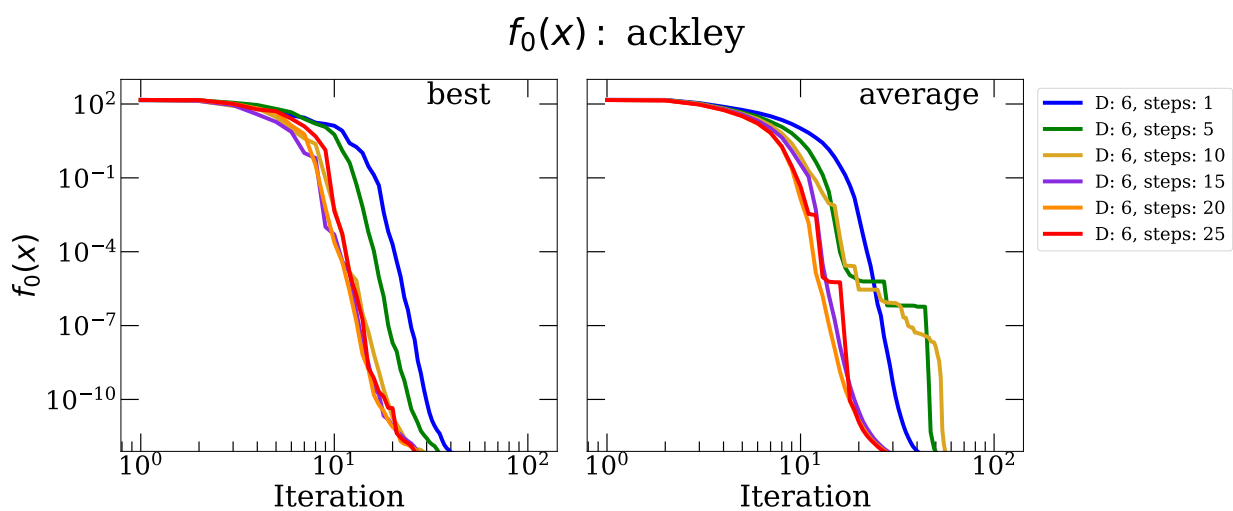


Figure 3. Convergence of the ackley function for various number of steps, 1D-SOA algorithm.

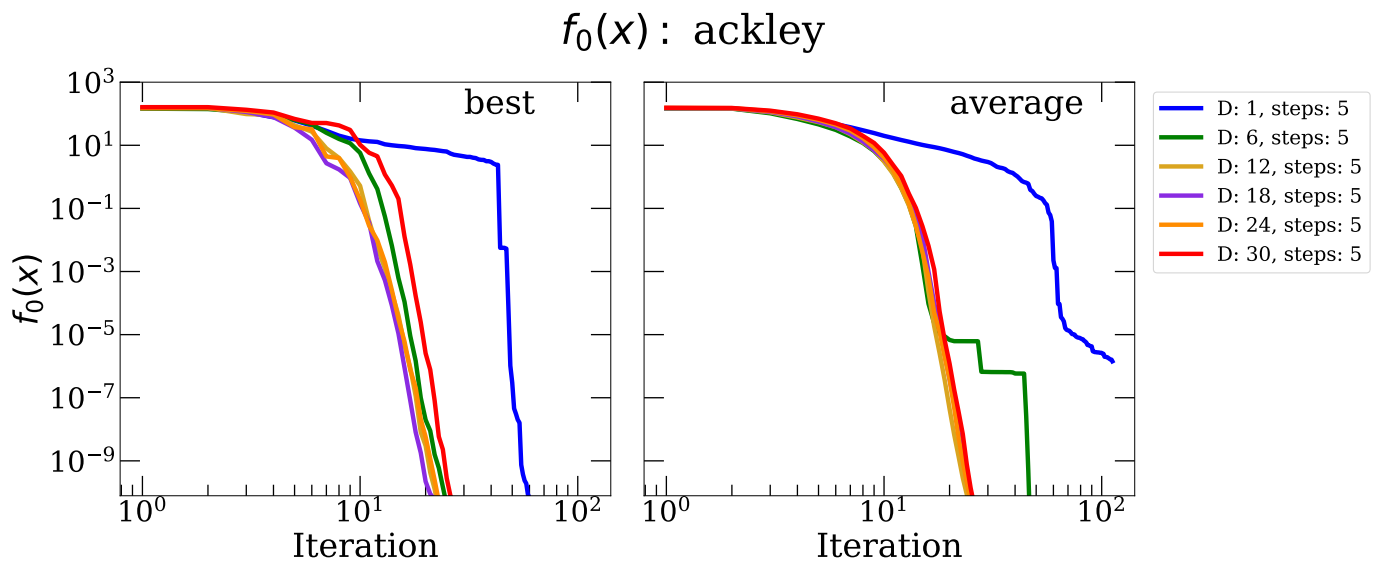


Figure 4. Convergence of the ackley function for various search directions, 1D-SOA algorithm.

4.1. Results for 2D Experiments

The complete results of this section are presented in the Appendix B, Tables A1–A3, that show the value that reaches the best solution out of the 100 runs, the averaged value reached, and the standard deviation for all the algorithms. The algorithm with the best performance has the darker corresponding value. If two algorithms have the same value, the one with faster convergence in the plots is selected. Additionally, two statistical tests were performed and included in these tables, the *t*-test [23], and the alternative to the Mann–Witney U Test, suggested by Mark Wineberg and Steffen Christensen, that is a non-parametric test based on the ranking of the results [54]. According to the non-parametric tests performed, most of the freedom degrees are close to 100, and the *t* statistic ranges from 3.3 to 49, which, from the Tables given in [54], indicates that the probability that the null hypotheses is wrong is over 99.8%; this implies that our results can be trusted. A summary of the results obtained for dimension 2 is presented in Table 3. In this table, we can observe that the best performance for the best solution was obtained for the 1D-SOA, while for the mean of the solutions, the best performance was achieved by the FWA. The convergence of the algorithms 1D-SOA, FWA, ZOA, AEO, MPSO, and CHSO for some functions is shown in Figure 5.

Table 3. Performance of the algorithms 2D.

	1D-SOA	FWA	ZOA	AEO	MPSO	CHSO
No. of best values	13 (43.3%)	8 (26.6%)	0	3 (10%)	0	6 (20%)
No. of mean values	10 (33.3%)	16 (53.3%)	0	4 (13.3%)	0	0

From the figures, it was observed that there exist two different behaviors of the algorithms depending on the function. For the first kind, all the algorithms converge at a similar rate, this behavior is observed for most of the functions, in particular, the functions: f_0 , f_1 , f_3 , f_4 , f_5 , f_9 , f_{10} , f_{14} , f_{15} , f_{16} , f_{17} , f_{18} , f_{20} , f_{21} , f_{22} , f_{23} , f_{24} , f_{25} , f_{26} , f_{27} , and f_{29} . For the second kind, the convergence is very slow for all the algorithms, except for one or two algorithms; this happens for functions f_2 , f_6 , f_7 , f_8 , f_{11} , f_{12} , f_{13} , f_{19} , and f_{28} .

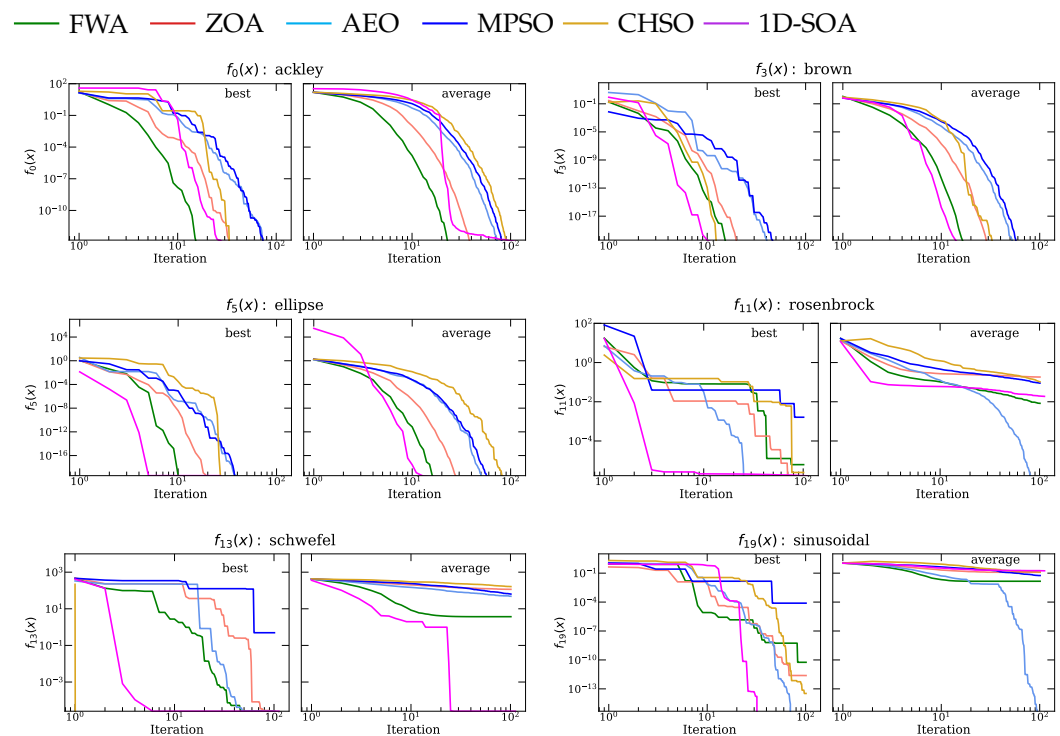


Figure 5. Convergence of the FWA, ZOA, AEO, MPSO, CHSO, and 1D-SOA algorithms for various functions in 2D.

4.2. Results for 30D

The complete results of this section are presented in the Appendix B, Tables A4–A6, that show the value that reaches the best solution out of the 100 runs, the averaged value reached, and the standard deviation for all the algorithms. As in the previous case (2D), the two statistical tests were performed and included in the tables. According to the non-parametric tests performed, most of the freedom degrees are close to or larger than 100, and the t statistic ranges from 3.7 to 52, which, from Tables [54], indicates that the probability that the null hypothesis is wrong is over 99.8%; this implies that our results can be trusted. A summary of the results obtained for dimension 30 is presented in Table 4. This table shows that the best performance for the best and mean solution was obtained for the 1D-SOA, followed by the FWA. The convergence of the algorithms 1D-SOA, FWA, ZOA, AEO, MPSO, and CHSO for some functions is shown in Figure 6.

Table 4. Performance of the algorithms 30D.

	1D-SOA	FWA	ZOA	AEO	MPSO	CHSO
No. of best values	21 (70%)	5 (16.6%)	1 (3.33%)	2 (6.66%)	0	1 (3.33%)
No. of mean values	19 (63.3%)	10 (33.3%)	0	1 (3.33%)	0	0

For 30D, the CHSO algorithm shows a general performance that is worse than the 2D case. As in the previous case, different behaviors of the algorithms depending on the function are observed. For most of the functions, the convergence is similar for all the algorithms, in particular, for the functions: f_3 , f_6 , f_9 , f_{10} , f_{12} , f_{20} , f_{21} , f_{22} , f_{23} , f_{25} , and f_{26} . There is also a case where the 1D-SOA, FWA, ZOA, AEO, and MPSO present similar performances, but the CHSO fails to converge, the functions with this behavior are: f_0 , f_1 , f_2 , f_4 , f_5 , f_{14} , f_{15} , f_{16} , f_{17} , f_{24} , and f_{26} . Finally, there is a case where only one or two functions converge or converge faster than the rest, this happens for functions: f_7 , f_8 , f_{11} , f_{13} , f_{18} , f_{19} , and f_{27} . For functions f_{28} and f_{29} , 1D-SOA does not converge. However, for functions f_7 , f_8 , f_{11} , and f_{13} only 1D-SOA converges.

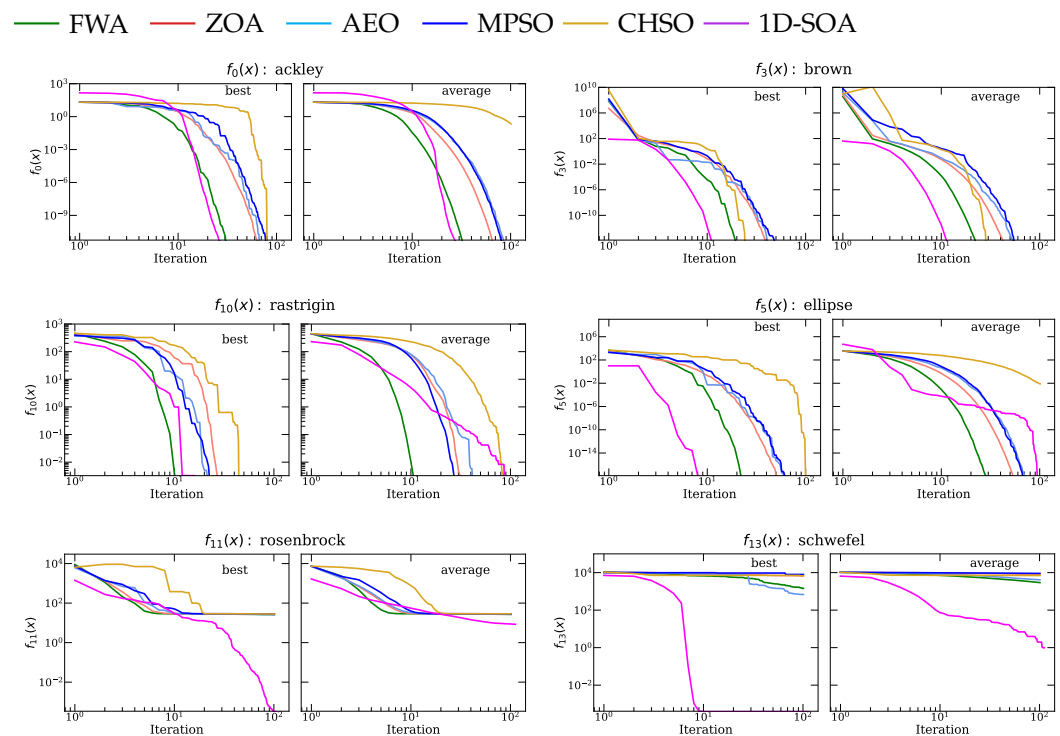


Figure 6. Convergence of the FWA, ZOA, AEO, MPSO, CHSO, and 1D-SOA algorithms for various functions in 30D.

4.3. Experiments with Functions with the Optimal Solution Not in Zero

The minimal value of the functions studied in the previous sections have a minimum $f(\vec{x}) = 0$ located in $\vec{x} = [0, \dots, 0]$. However, some of them have the optimum in a different position. In this section, the functions with a non-zero optimum are studied; they are presented in Table A11, together with their optimal value. The functions f_7 and f_8 can be generalized to D dimensions and they are studied in the previous section; only the minimum value was shifted to zero, therefore we only show their convergence plots if they are not shifted towards zero in Figure 7. The rest of the functions are only defined in 2D. The results are presented in Table A7. For some of the functions studied in this section, the CHSO algorithm showed problems computing the solution, therefore, the studied algorithms are 1D-SOA, FWA, ZOA, AEO, and MPSO. The convergence plots are presented in Figure 8.

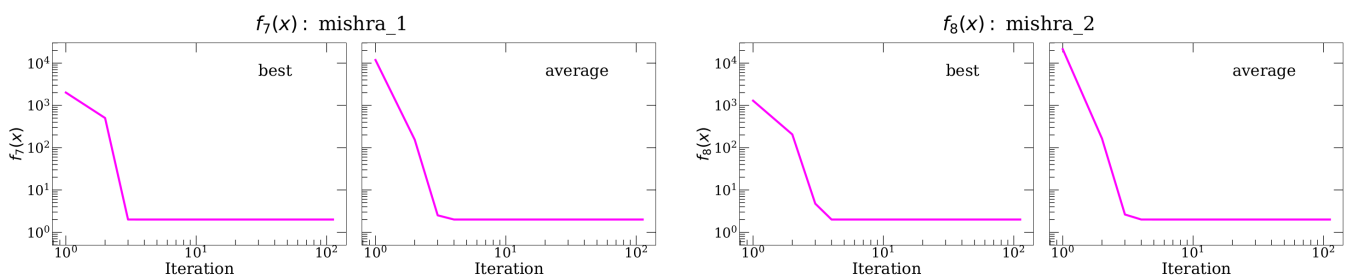


Figure 7. Convergence of the 1D-SOA algorithms for functions with a non-zero optimum for functions f_7 and f_8 .

Regarding the best value, for these functions, most of the results are similar for all the algorithms, except for f_{32} and f_{36} , where FWA for f_{32} and 1D-SOA for f_{36} converge slightly faster than the rest to the optimum. For the Mean value, FWA reaches values closer to the optimum.

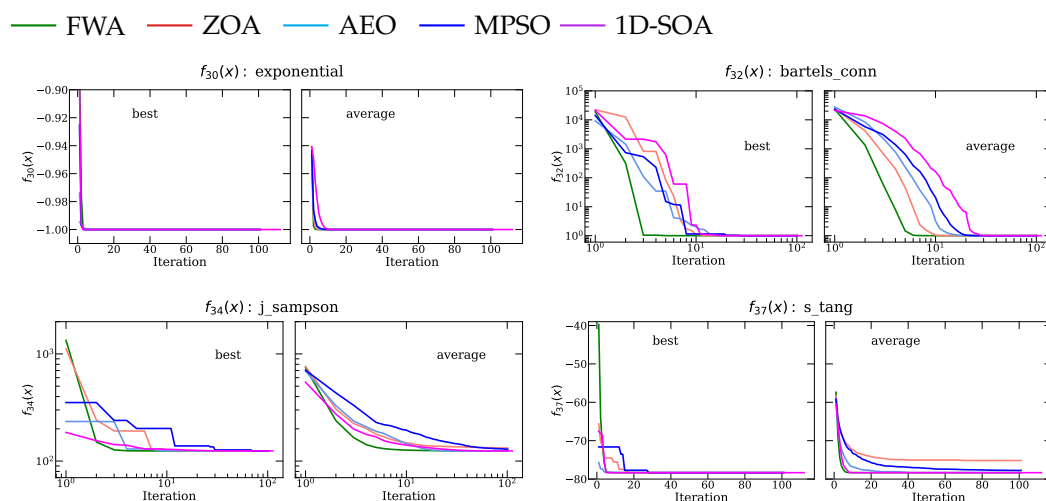


Figure 8. Convergence of the FWA, ZOA, AEO, MPSO, and 1D-SOA algorithms for various functions with a non-zero optimum in 2D.

5. Conclusions

This work introduces the one-dimensional search optimization algorithm (1D-SOA) for solving optimization problems. This algorithm optimizes an initial population over a randomly selected dimension. It selects the search direction by moving symmetrically in the one-dimensional subspace (1D-S) created by the selected dimension. After the direction of the search is selected, the individuals move asymmetrically to find the local optima in the 1D-S. The algorithm includes exploitation by diminishing the size of the step to perform a local search. It also performs exploration by recombining the particles. To study the performance and efficiency of the proposed algorithm, it was compared against another 11 algorithms for 30 benchmark functions for various dimensions.

The comparison was made for two cases: (1) the best values reached by the algorithms, and (2) the mean performance out of 100 runs. For the best value, the best performance was observed for the 1D-SOA, for which 43.3% of the functions were in 2D, 70% in 30D; followed by the FWA with 26.6% in 2D, 16.6% in 30D; the CHSO with 20% in 2D; and the AEO with 6.6% in 30D. For the mean convergence, FWA showed a better performance with 53.3%, followed by the 1D-SOA with 33.3% of the functions in 2D. In 30D, the best performance was observed by the 1D-SOA algorithm with 63.3%, followed by the FWA with 33.3%. A *t*-test and a non-parametric test were performed to check the validity of the results.

The functions with non-zero optimum were studied for 2D, and the results were similar for the best solution except for the f_{32} and f_{36} functions, where the best performance was obtained with the FWA and the 1D-SOA algorithms. The FWA showed the best performance for the mean solutions.

From the results, it can be concluded that the best performance was achieved with the proposed algorithm (1D-SOA) in 30D, followed by the fireworks algorithm (FWA). For most of the functions, the convergence was similar for 2D. However, for larger dimensions, the chicken swarm optimization (CHSO) algorithm worsens its performance, while 1D-SOA showed a better performance. Furthermore, for some functions, 1D-SOA was the only algorithm that converged in large dimensions.

A fixed set of parameters was chosen for 1D-SOA to compare it against the other algorithms. However, these parameters can be adapted for diverse optimization problems. It is suggested to carry out an analysis of the parameters to find the most adequate ones for a given problem.

Author Contributions: Conceptualization, G.B.D.-C.; methodology, G.B.D.-C. and R.L.-G.; software, G.B.D.-C. and R.L.-G.; investigation, R.L.-G.; writing—original draft preparation, G.B.D.-C.; writing—review and editing, R.L.-G.; supervision, R.L.-G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Institute Politécnico Nacional through the project SIP20220354.

Data Availability Statement: Not applicable

Acknowledgments: We thank CONACyT for their partial support of the present work.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MDPI	Multidisciplinary Digital Publishing Institute
1D-SOA	One-Dimensional Search Algorithm
ABA	Artificial Bee Algorithm
BA	Bat Algorithm
BFO	Bacterial Foraging Optimization
CA	Cat Swarm Optimization
CHSO	Chicken Swarm Optimization
CU	Cuckoo Search Optimization
FA	Firefly algorithm
FWA	Fireworks Algorithm
GSA	Gravitational Search Algorithm
SSA	Social Spider Algorithm
PSO	Particle Swarm Optimization
MPSO	Mean Particle Swarm Optimization
AEO	Artificial ecosystem-based optimization
JF	Jellyfish-inspired metaheuristic
CGO	Chaos Game Optimizer (CGO)
ZOA	Zebra Optimization Algorithm (ZOA)
CSA	Chameleon Swarm Algorithm
SOA	Serval Optimization Algorithm
OWB	Optimization based on walrus behavior
COA	Coronavirus Optimization Algorithm
EA4EIG	Eigen Crossover in Cooperative Model of Evolutionary Algorithms
APGSK	Gaining–Sharing Knowledge-Based Algorithm With Adaptive Parameters
IMODE	Improving Multi-Objective Differential Evolution Algorithm
D	Dimension
N	Number of individuals
\mathcal{G}	Generation
$Max_{\mathcal{G}}$	Maximum number of generations
ϵ	Difference between previous and actual point
It_{ϵ}	Maximum allowed difference between previous and actual point
Max_{ϵ}	Maximum allowed difference between previous and actual point
it_{NI}	Counter for non improvement of the solution
Max_{NI}	Maximum allowed number of non improvements of the solution
x_{min}	Minimum allowed value of the function
x_{max}	Maximum allowed value of the function
C_{neigh}	Neighborhood parameter

C_{neigh_l}	Local neighborhood parameter
Δ	Search step
D_{test}	Random test directions to select the direction with the largest improvement
$steps$	Number of steps given in each test direction
N_{px}	Percentage of particles on which to perform semi cross
P_{sx}	Probability of semi cross
P_{ns}	Probability of neighborhood search
P_{dim}	Number of search dimensions
Nr	Percentage of particles to recombine

Appendix A. Experiments with Diverse Parameters for the Studied Algorithms

A subset of functions and diverse parameters were selected in order to test and find the best internal parameters for the MPSO, CHOA, and FWA. For MPSO, the studied parameters are values in the range proposed by the authors of references [49,53]. The parameters varied are the $c1$ and $c2$, which are the ratios between the cognitive and social components. The results are presented in Figure A1 for 30D. It is observed that the best values are $c1 = 2$ and $c2 = 2$, presented in Table 2 and used in this work.

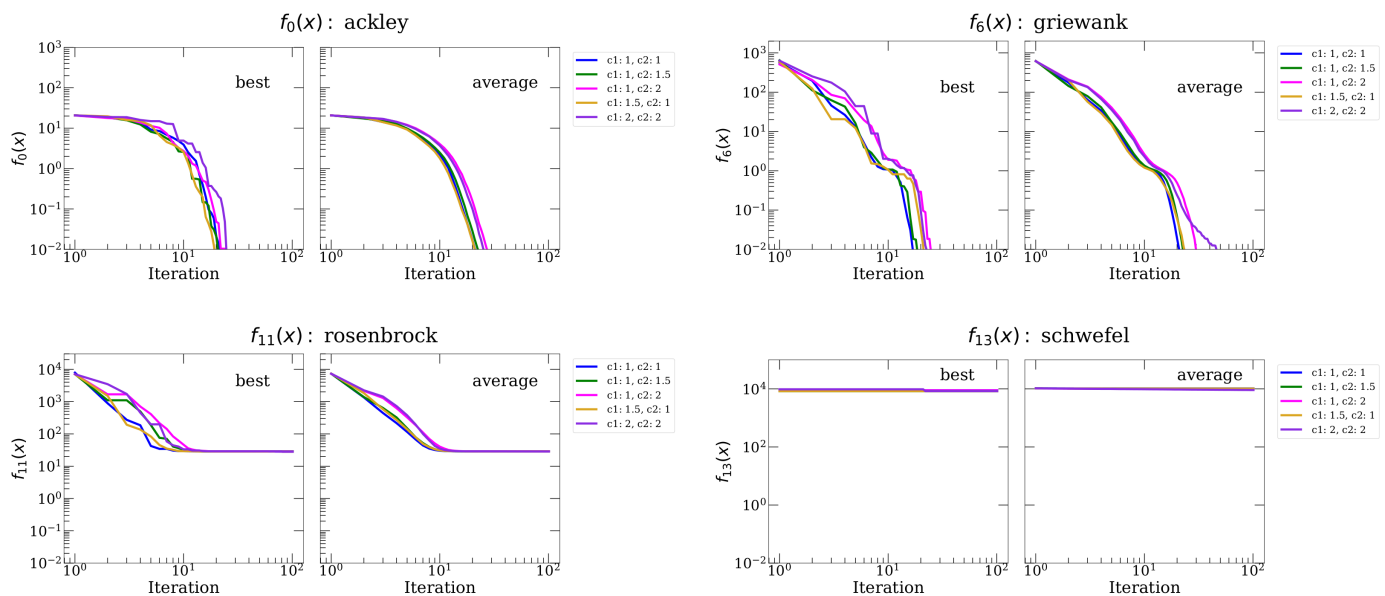


Figure A1. Convergence of the MPSO algorithm for diverse parameters 30D.

For the CHSO algorithm, the studied parameters are G , the time to upgrade the relationships, and FL , which shows that, whether a chicken follows the mother or not, in the range proposed by the authors of reference [49] and reference [55], the results are presented in Figure A2. It is observed that, for the mean, there is not large variation when the parameters are changed. Therefore, the parameters used are, again, the default of the SwarmPackagePy [49], as shown in Table 2.

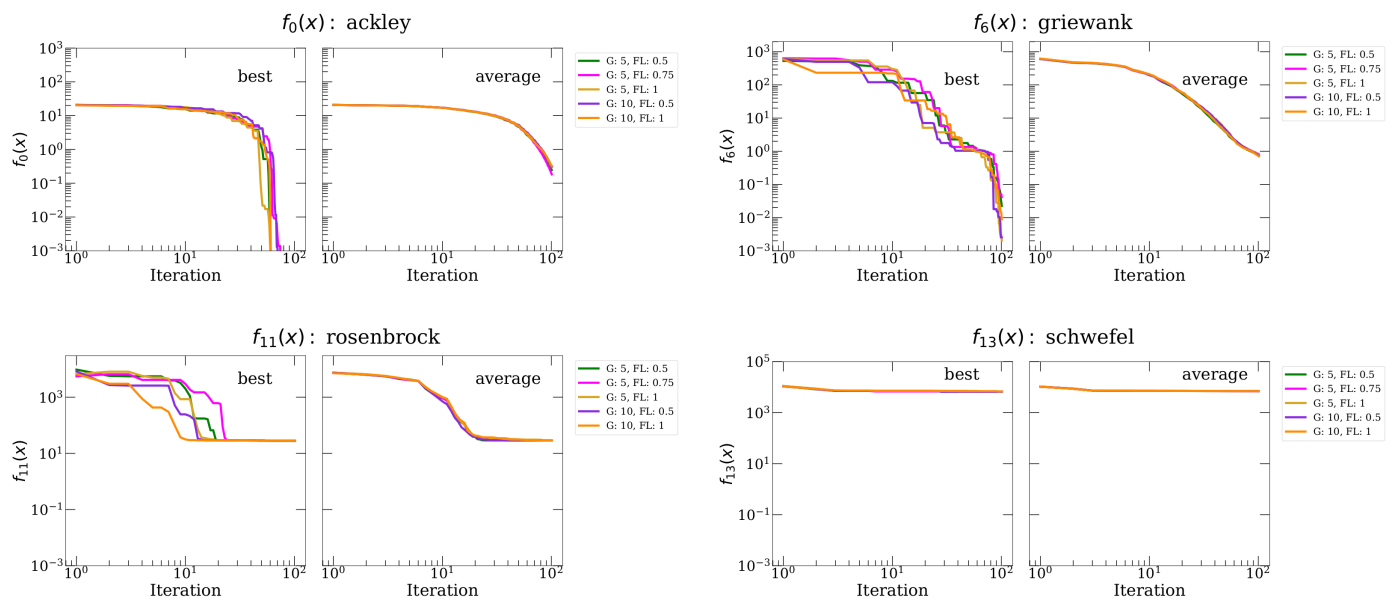


Figure A2. Convergence of the CHSO algorithm for diverse parameters for 30D.

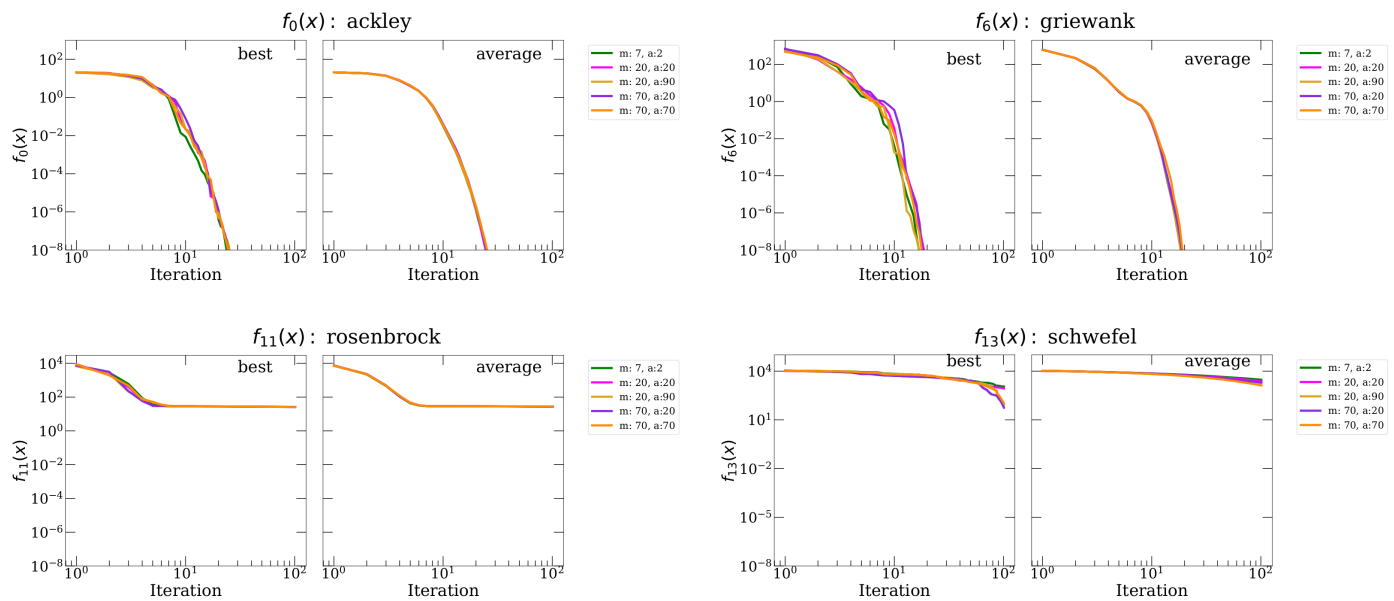


Figure A3. Convergence of the FWA algorithm for diverse parameters for 30D.

For the FWA algorithm, the $m1$ and $m2$ parameters were varied in the ranges given by the authors of references [49,55]. The results are shown in Figure A3 for 30D, where it can be observed that there is no large difference for different parameters, therefore the default of the SwarmPackagePy 1.0.0a5 is used; they are presented in Table 2.

Appendix B. Results for the Selected Algorithms

Table A1. Experimental results of various algorithms and benchmark functions for 2D.

		Best	Mean	Std	d	t	d_{np}	t_{np}
f_0	1D-SOA	0.0	$4.9e^{-12}$	$1.4e^{-11}$				
	FWA	$4.4e^{-16}$	$5.2e^{-16}$	$5.0e^{-16}$	99	3.5	101	8.2
	ZOA	$1.1e^{-11}$	$7.8e^{-9}$	$1.8e^{-8}$	99	4.2	101	8.2
	AEO	$9.1e^{-8}$	$2.2e^{-4}$	$6.3e^{-4}$	99	3.4	101	8.2
	MPSO	$5.2e^{-6}$	$9.4e^{-4}$	$1.4e^{-3}$	99	6.9	101	8.2
	CHSO	$4.4e^{-16}$	$6.6e^{-3}$	$1.7e^{-2}$	99	3.8	101	8.2
f_1	1D-SOA	$5.1e^{-11}$	$7.4e^{-6}$	$6.7e^{-5}$				
	FWA	$2.2e^{-10}$	$5.0e^{-6}$	$1.3e^{-5}$	106	$3.6e^{-1}$	197	8.1
	ZOA	$7.5e^{-6}$	$5.8e^{-3}$	$1.0e^{-2}$	99	5.7	197	8.1
	AEO	$1.0e^{-3}$	$1.5e^{-1}$	$2.4e^{-1}$	99	6.2	197	8.1
	MPSO	$5.4e^{-3}$	$5.3e^{-1}$	$5.9e^{-1}$	99	9.0	197	8.1
	CHSO	$2.7e^{-3}$	3.5	4.4	99	7.9	197	8.1
f_2	1D-SOA	$2.8e^{-13}$	$5.5e^{-10}$	$1.4e^{-9}$				
	FWA	$7.5e^{-16}$	$8.0e^{-8}$	$3.7e^{-7}$	99	2.1	184	8.8
	ZOA	$7.5e^{-7}$	$2.2e^{-3}$	$6.5e^{-3}$	99	3.3	184	8.8
	AEO	$3.1e^{-5}$	$2.5e^{-2}$	$6.6e^{-2}$	99	3.8	184	8.8
	MPSO	$2.6e^{-5}$	$6.8e^{-2}$	$1.3e^{-1}$	99	5.2	184	8.8
	CHSO	$3.3e^{-9}$	$9.3e^{-2}$	$2.1e^{-1}$	99	4.4	184	8.8
f_3	1D-SOA	$2.5e^{-23}$	$2.1e^{-16}$	$9.4e^{-16}$				
	FWA	$7.8e^{-23}$	$1.3e^{-12}$	$4.6e^{-12}$	99	2.9	153	4.9
	ZOA	$2.1e^{-11}$	$9.6e^{-8}$	$2.8e^{-7}$	98	3.4	153	4.9
	AEO	$3.8e^{-9}$	$6.7e^{-5}$	$1.8e^{-4}$	99	3.8	153	4.9
	MPSO	$5.7e^{-8}$	$3.1e^{-4}$	$6.2e^{-4}$	99	5.0	153	4.9
	CHSO	0.0	$4.7e^{-4}$	$3.3e^{-3}$	99	1.4	153	4.9
f_4	1D-SOA	$9.2e^{-21}$	$2.3e^{-3}$	$2.1e^{-2}$				
	FWA	$1.0e^{-13}$	$1.6e^{-3}$	$1.3e^{-2}$	168	$2.7e^{-1}$	196	8.9
	ZOA	$5.3e^{-4}$	7.7	$2.1e^1$	99	3.7	196	8.9
	AEO	$2.5e^{-2}$	$5.0e^2$	$1.2e^3$	99	4.0	196	8.9
	MPSO	$3.5e^{-1}$	$6.1e^4$	$3.4e^5$	99	1.8	196	8.9
	CHSO	$3.9e^{-1}$	$1.1e^6$	$5.1e^6$	99	2.1	196	8.9
f_5	1D-SOA	$2.2e^{-26}$	$2.9e^{-14}$	$2.2e^{-13}$				
	FWA	$4.8e^{-18}$	$1.6e^{-9}$	$9.8e^{-9}$	99	1.6	177	$1.5e^1$
	ZOA	$4.3e^{-13}$	$5.9e^{-6}$	$1.3e^{-5}$	99	4.6	177	$1.5e^1$
	AEO	$2.8e^{-8}$	$2.2e^{-3}$	$1.8e^{-2}$	99	1.2	177	$1.5e^1$
	MPSO	$2.8e^{-6}$	$3.9e^{-3}$	$1.0e^{-2}$	99	3.8	177	$1.5e^1$
	CHSO	$8.8e^{-7}$	$2.3e^{-2}$	$4.9e^{-2}$	99	4.6	177	$1.5e^1$
f_6	1D-SOA	0.0	$5.5e^{-3}$	$5.9e^{-3}$				
	119	9.0						
	FWA	0.0	$1.5e^{-4}$	$1.0e^{-3}$	105	9.0	119	9.0
	ZOA	$5.4e^{-12}$	$1.5e^{-2}$	$1.4e^{-2}$	130	6.2	119	9.0
	AEO	$1.0e^{-5}$	$8.5e^{-2}$	$7.2e^{-2}$	100	$1.1e^1$	119	9.0
	MPSO	$8.2e^{-8}$	$1.6e^{-1}$	$1.1e^{-1}$	99	$1.4e^1$	119	9.0
	CHSO	$5.7e^{-4}$	$2.2e^{-1}$	$2.3e^{-1}$	99	9.6	119	9.0
f_7	1D-SOA	$3.2e^{-12}$	$6.6e^{-3}$	$5.0e^{-2}$				
	FWA	$1.3e^{-5}$	$1.0e^{-2}$	$1.3e^{-2}$	111	$6.9e^{-1}$	99	1.0
	ZOA	0.0	$2.9e^{-1}$	$6.2e^{-1}$	100	4.5	99	1.0
	AEO	0.0	0.0	0.0	98	1.3	99	1.0
	MPSO	$1.3e^{-3}$	$9.9e^{-2}$	$1.1e^{-1}$	139	7.7	99	1.0
	CHSO	0.0	$1.6e^{-1}$	$5.2e^{-1}$	100	2.9	99	1.0
f_8	1D-SOA	$2.1e^{-12}$	$5.4e^{-9}$	$2.8e^{-8}$				
	FWA	$1.6e^{-3}$	$1.3e^{-2}$	$9.7e^{-3}$	99	$1.3e^1$	99	3.6
	ZOA	0.0	$5.6e^{-1}$	$5.0e^{-1}$	99	$1.1e^1$	99	3.6
	AEO	0.0	0.0	0.0	99	1.9	99	3.6
	MPSO	$4.3e^{-2}$	$3.8e^{-1}$	$2.2e^{-1}$	99	$1.7e^1$	99	3.6
	CHSO	0.0	$1.5e^{-1}$	$3.6e^{-1}$	99	4.3	99	3.6

Table A2. Experimental results of various algorithms and benchmark functions for 2D.

		Best	Mean	Std	d	t	d_{np}	t_{np}
f_9	1D-SOA	$1.9e^{-48}$	$1.6e^{-35}$	$1.5e^{-34}$				
	FWA	$1.1e^{-49}$	$2.6e^{-27}$	$1.6e^{-26}$	99	1.6	153	1.4
	ZOA	$3.4e^{-26}$	$9.0e^{-16}$	$7.1e^{-15}$	99	1.3	153	1.4
	AEO	$4.0e^{-23}$	$3.9e^{-10}$	$2.4e^{-9}$	99	1.6	153	1.4
	MPSO	$1.5e^{-18}$	$9.8e^{-9}$	$3.9e^{-8}$	99	2.5	153	1.4
	CHSO	$1.2e^{-54}$	$1.4e^{-7}$	$1.3e^{-6}$	99	1.0	153	1.4
f_{10}	1D-SOA	0.0	$5.3e^{-17}$	$3.0e^{-16}$				
	FWA	0.0	$1.4e^{-16}$	$9.9e^{-16}$	117	$8.5e^{-1}$	192	1.3
	ZOA	$1.5e^{-12}$	$9.8e^{-2}$	$3.0e^{-1}$	99	3.3	192	1.3
	AEO	$1.9e^{-10}$	$5.6e^{-1}$	$9.1e^{-1}$	99	6.2	192	1.3
	MPSO	$1.2e^{-6}$	1.3	1.5	98	8.7	192	1.3
	CHSO	$3.6e^{-15}$	1.4	2.0	99	7.0	192	1.3
f_{11}	1D-SOA	$1.7e^{-6}$	$2.0e^{-2}$	$2.7e^{-2}$				
	FWA	$6.2e^{-6}$	$7.2e^{-3}$	$1.2e^{-2}$	137	4.4	197	$6.4e^{-1}$
	ZOA	$5.2e^{-8}$	$1.8e^{-1}$	$2.1e^{-1}$	102	7.4	197	$6.4e^{-1}$
	AEO	$1.9e^{-16}$	$4.2e^{-8}$	$2.8e^{-7}$	99	7.5	197	$6.4e^{-1}$
	MPSO	$1.6e^{-3}$	$9.2e^{-2}$	$8.1e^{-2}$	120	8.4	197	$6.4e^{-1}$
	CHSO	$2.5e^{-6}$	$1.0e^{-1}$	$1.6e^{-1}$	104	5.1	197	$6.4e^{-1}$
f_{12}	1D-SOA	$1.0e^{-1}$	$9.4e^{-1}$	$5.5e^{-1}$				
	FWA	$1.8e^{-91}$	$1.3e^{-69}$	$1.3e^{-68}$	99	$1.7e^1$	99	$1.7e^1$
	ZOA	$1.3e^{-38}$	$5.0e^{-3}$	$2.2e^{-2}$	99	$1.7e^1$	99	$1.7e^1$
	AEO	$9.5e^{-24}$	$1.0e^{-3}$	$9.9e^{-3}$	99	$1.7e^1$	99	$1.7e^1$
	MPSO	$6.6e^{-21}$	$1.9e^{-2}$	$3.5e^{-2}$	99	$1.7e^1$	99	$1.7e^1$
	CHSO	$9.5e^{-17}$	$3.1e^{-2}$	$4.5e^{-2}$	100	$1.7e^1$	99	$1.7e^1$
f_{13}	1D-SOA	$2.5e^{-5}$	2.4	$1.7e^1$				
	FWA	$5.4e^{-4}$	$1.2e^1$	$4.2e^1$	128	2.1	122	8.3
	ZOA	$9.1e^{-2}$	$2.0e^2$	$1.2e^2$	103	$1.7e^1$	122	8.3
	AEO	$1.2e^{-1}$	$1.5e^2$	$1.1e^2$	103	$1.4e^1$	122	8.3
	MPSO	4.2	$2.5e^2$	$1.2e^2$	102	$2.1e^1$	122	8.3
	CHSO	$-4.1e^2$	$3.2e^2$	$1.3e^2$	102	$2.4e^1$	122	8.3
f_{14}	1D-SOA	$3.5e^{-12}$	$7.2e^{-9}$	$1.2e^{-8}$				
	FWA	$4.7e^{-12}$	$2.7e^{-6}$	$8.2e^{-6}$	99	3.3	175	$1.1e^1$
	ZOA	$2.9e^{-5}$	$3.3e^{-3}$	$3.6e^{-3}$	99	9.0	175	$1.1e^1$
	AEO	$5.5e^{-5}$	$1.1e^{-1}$	$1.2e^{-1}$	99	9.1	175	$1.1e^1$
	MPSO	$6.7e^{-3}$	$4.6e^{-1}$	$5.3e^{-1}$	99	8.7	175	$1.1e^1$
	CHSO	$4.1e^{-3}$	2.3	3.7	98	6.4	175	$1.1e^1$
f_{15}	1D-SOA	$1.3e^{-9}$	$1.0e^{-6}$	$2.9e^{-6}$				
	FWA	$7.3e^{-20}$	$9.8e^{-13}$	$5.9e^{-12}$	99	3.5	126	3.8
	ZOA	$1.8e^{-10}$	$7.3e^{-7}$	$1.2e^{-6}$	131	$9.1e^{-1}$	126	3.8
	AEO	$3.5e^{-6}$	$1.7e^{-3}$	$6.1e^{-3}$	99	2.7	126	3.8
	MPSO	$1.7e^{-5}$	$7.8e^{-3}$	$1.2e^{-2}$	99	6.2	126	3.8
	CHSO	$9.3e^{-6}$	$2.6e^{-1}$	$6.7e^{-1}$	99	3.9	126	3.8
f_{16}	1D-SOA	$2.4e^{-14}$	$7.3e^{-11}$	$1.5e^{-10}$				
	FWA	$1.4e^{-21}$	$5.1e^{-13}$	$2.7e^{-12}$	99	4.8	197	$1.9e^1$
	ZOA	$9.1e^{-10}$	$7.0e^{-7}$	$2.0e^{-6}$	99	3.5	197	$1.9e^1$
	AEO	$1.2e^{-6}$	$1.2e^{-3}$	$2.5e^{-3}$	99	4.9	197	$1.9e^1$
	MPSO	$4.9e^{-7}$	$1.8e^{-2}$	$3.4e^{-2}$	99	5.2	197	$1.9e^1$
	CHSO	$1.7e^{-6}$	$1.3e^{-1}$	$3.2e^{-1}$	99	4.2	197	$1.9e^1$
f_{17}	1D-SOA	$4.8e^{-100}$	$4.8e^{-72}$	$4.6e^{-71}$				
	FWA	$5.1e^{-107}$	$1.9e^{-49}$	$1.9e^{-48}$	98	1.0	171	3.6
	ZOA	$2.2e^{-53}$	$1.3e^{-28}$	$8.7e^{-28}$	99	1.5	171	3.6
	AEO	$3.2e^{-50}$	$8.0e^{-13}$	$7.9e^{-12}$	99	1.0	171	3.6
	MPSO	$1.8e^{-33}$	$9.5e^{-8}$	$6.6e^{-7}$	98	1.5	171	3.6
	CHSO	$4.1e^{-60}$	$2.6e^3$	$2.2e^4$	98	1.2	171	3.6
f_{18}	1D-SOA	$4.7e^{-8}$	1.9	5.2				
	FWA	$8.9e^{-14}$	$2.5e^{-4}$	$1.1e^{-3}$	99	3.6	144	1.2
	ZOA	$1.4e^{-6}$	$1.8e^{-1}$	$3.6e^{-1}$	99	3.2	144	1.2
	AEO	$4.0e^{-4}$	4.1	$1.2e^1$	133	1.7	144	1.2
	MPSO	$5.9e^{-3}$	9.8	$2.4e^1$	107	3.2	144	1.2
	CHSO	0.0	$7.4e^1$	$1.4e^2$	99	5.1	144	1.2

Table A3. Experimental results of various algorithms and benchmark functions for 2D.

		Best	Mean	Std	d	t	d_{np}	t_{np}
f_{19}	1D-SOA	0.0	$1.6e^{-1}$	$3.2e^{-1}$				
	FWA	$5.8e^{-11}$	$1.6e^{-2}$	$1.1e^{-1}$	122	4.2	176	$1.5e^1$
	ZOA	$2.4e^{-12}$	$1.2e^{-1}$	$2.9e^{-1}$	195	$9.3e^{-1}$	176	$1.5e^1$
	AEO	0.0	$4.4e^{-17}$	$4.4e^{-16}$	98	5.0	176	$1.5e^1$
	MPSO	$7.7e^{-5}$	$5.9e^{-2}$	$5.8e^{-2}$	105	3.1	176	$1.5e^1$
	CHSO	$3.3e^{-14}$	$1.3e^{-1}$	$4.7e^{-1}$	174	$4.7e^{-1}$	176	$1.5e^1$
f_{20}	1D-SOA	$4.5e^{-24}$	$2.3e^{-10}$	$1.5e^{-9}$				
	FWA	$2.6e^{-24}$	$1.4e^{-13}$	$5.3e^{-13}$	99	1.5	190	6.3
	ZOA	$3.8e^{-12}$	$1.8e^{-7}$	$5.0e^{-7}$	99	3.5	190	6.3
	AEO	$6.3e^{-12}$	$1.3e^{-4}$	$3.4e^{-4}$	99	3.9	190	6.3
	MPSO	$4.4e^{-8}$	$9.9e^{-4}$	$2.8e^{-3}$	99	3.6	190	6.3
	CHSO	$1.2e^{-10}$	$1.5e^{-2}$	$5.7e^{-2}$	99	2.6	190	6.3
f_{21}	1D-SOA	$4.1e^{-23}$	$2.9e^{-17}$					
	FWA	$4.8e^{-22}$	$1.4e^{-12}$	$1.1e^{-11}$	99	1.3	176	8.6
	ZOA	$1.7e^{-11}$	$7.2e^{-7}$	$2.1e^{-6}$	98	3.5	176	8.6
	AEO	$2.0e^{-10}$	$3.9e^{-4}$	$1.1e^{-3}$	99	3.5	176	8.6
	MPSO	$2.1e^{-8}$	$5.1e^{-3}$	$1.1e^{-2}$	99	4.5	176	8.6
	CHSO	$8.0e^{-17}$	$1.3e^{-1}$	$4.4e^{-1}$	98	3.0	176	8.6
f_{22}	1D-SOA	$1.2e^{-17}$	$3.7e^{-4}$	$2.1e^{-3}$				
	FWA	$2.5e^{-14}$	$9.0e^{-8}$	$3.4e^{-7}$	99	1.7	192	4.8
	ZOA	$7.3e^{-9}$	$2.4e^{-5}$	$9.1e^{-5}$	99	1.6	192	4.8
	AEO	$3.8e^{-8}$	$2.3e^{-4}$	$4.5e^{-4}$	107	$6.2e^{-1}$	192	4.8
	MPSO	$2.9e^{-7}$	$7.5e^{-4}$	$1.3e^{-3}$	166	1.5	192	4.8
	CHSO	$2.4e^{-13}$	$1.6e^{-4}$	$9.4e^{-4}$	137	$8.8e^{-1}$	192	4.8
f_{23}	1D-SOA	0.0	0.0	0.0				
	FWA	0.0	0.0	0.0				
	ZOA	0.0	0.0	0.0				
	AEO	0.0	0.0	0.0				
	MPSO	0.0	0.0	0.0				
	CHSO	0.0	0.0	0.0				
f_{24}	1D-SOA	$5.5e^{-23}$	$2.2e^1$	$1.1e^2$				
	FWA	$3.7e^{-23}$	$6.0e^{-8}$	$5.0e^{-7}$	99	2.1	169	4.7
	ZOA	$8.4e^{-7}$	$1.5e^{-2}$	$5.3e^{-2}$	99	2.1	169	4.7
	AEO	$2.6e^{-5}$	$1.5e^1$	$3.7e^1$	123	$6.9e^{-1}$	169	4.7
	MPSO	$2.0e^{-3}$	$2.1e^2$	$6.9e^2$	103	2.7	169	4.7
	CHSO	$2.2e^{-5}$	$4.0e^3$	$2.5e^4$	99	1.6	169	4.7
f_{25}	1D-SOA	0.0	$1.8e^{-3}$	$1.2e^{-2}$				
	FWA	0.0	$3.1e^{-8}$	$1.5e^{-7}$	99	1.4	195	7.6
	ZOA	$7.5e^{-8}$	$3.2e^{-2}$	$5.5e^{-2}$	109	5.4	195	7.6
	AEO	$1.3e^{-5}$	$1.1e^{-1}$	$1.0e^{-1}$	101	$1.0e^1$	195	7.6
	MPSO	$8.3e^{-6}$	$1.9e^{-1}$	$1.4e^{-1}$	100	$1.3e^1$	195	7.6
	CHSO	0.0	$7.6e^{-2}$	$1.2e^{-1}$	101	6.3	195	7.6
f_{26}	1D-SOA	$3.9e^{-14}$	$9.1e^{-4}$	$5.5e^{-3}$				
	FWA	$6.0e^{-13}$	$1.5e^{-8}$	$4.8e^{-8}$	99	1.7	176	3.5
	ZOA	$6.0e^{-8}$	$3.2e^{-5}$	$9.3e^{-5}$	99	1.6	176	3.5
	AEO	$5.2e^{-6}$	$1.5e^{-3}$	$2.3e^{-3}$	133	$9.7e^{-1}$	176	3.5
	MPSO	$3.6e^{-5}$	$8.8e^{-3}$	$1.4e^{-2}$	128	5.3	176	3.5
	CHSO	$6.9e^{-11}$	$5.2e^{-2}$	$1.2e^{-1}$	99	4.3	176	3.5
f_{27}	1D-SOA	$1.1e^{-8}$	$1.8e^{-1}$	$1.6e^{-1}$				
	FWA	$5.4e^{-11}$	$1.2e^{-5}$	$4.3e^{-5}$	99	$1.1e^1$	180	6.1
	ZOA	$4.1e^{-5}$	$3.8e^{-2}$	$9.1e^{-2}$	154	7.6	180	6.1
	AEO	$6.8e^{-4}$	$1.3e^{-1}$	$1.5e^{-1}$	197	2.1	180	6.1
	MPSO	$1.4e^{-3}$	$1.4e^{-1}$	$1.2e^{-1}$	184	2.0	180	6.1
	CHSO	0.0	$1.6e^{-1}$	$1.6e^{-1}$	197	1.0	180	6.1
f_{28}	1D-SOA	$7.3e^{-11}$	$2.4e^{-1}$	$4.3e^{-1}$				
	FWA	$3.1e^{-11}$	$6.9e^{-7}$	$2.3e^{-6}$	99	5.6	174	2.9
	ZOA	-1.0	$-8.5e^{-1}$	$3.2e^{-1}$	182	$2.1e^1$	174	2.9
	AEO	-1.0	$-6.5e^{-1}$	$4.2e^{-1}$	197	$1.5e^1$	174	2.9
	MPSO	$6.4e^{-4}$	$5.6e^{-1}$	$3.9e^{-1}$	196	5.6	174	2.9
	CHSO	0.0	$4.5e^{-1}$	$4.3e^{-1}$	197	3.5	174	2.9
f_{29}	1D-SOA	$1.8e^{-13}$	$1.8e^{-1}$	$6.4e^{-1}$				
	FWA	$2.2e^{-22}$	$2.9e^{-12}$	$1.4e^{-11}$	99	2.8	181	8.6
	ZOA	$3.2e^{-12}$	$8.3e^{-7}$	$1.7e^{-6}$	99	2.8	181	8.6
	AEO	$1.2e^{-8}$	$7.6e^{-4}$	$2.2e^{-3}$	99	2.8	181	8.6
	MPSO	$3.5e^{-6}$	$3.0e^{-3}$	$7.8e^{-3}$	99	2.7	181	8.6
	CHSO	0.0	$4.4e^{-2}$	$1.2e^{-1}$	105	2.1	181	8.6

Table A4. Experimental results of various algorithms and benchmark functions for 30D.

		Best	Mean	Std	d	t	d_{np}	t_{np}
f_0	1D-SOA	$1.7e^{-12}$	$1.3e^{-11}$	$2.4e^{-11}$				
	FWA	$3.1e^{-12}$	$2.6e^{-9}$	$5.9e^{-9}$	99	4.4	194	$2.0e^1$
	ZOA	$8.3e^{-5}$	$5.3e^{-4}$	$3.2e^{-4}$	99	$1.7e^1$	194	$2.0e^1$
	AEO	$2.1e^{-7}$	$6.0e^{-3}$	$1.0e^{-2}$	99	5.9	194	$2.0e^1$
	MPSO	$7.1e^{-4}$	$9.2e^{-3}$	$1.5e^{-2}$	98	5.9	194	$2.0e^1$
	CHSO	$1.0e^{-1}$	$1.1e^1$	2.2	99	$5.0e^1$	194	$2.0e^1$
f_1	1D-SOA	$4.0e^{-12}$	$2.5e^{-11}$	$1.9e^{-11}$				
	FWA	$2.8e^{-9}$	$3.3e^{-6}$	$4.8e^{-6}$	99	6.8	197	$2.4e^1$
	ZOA	$4.0e^{-3}$	$2.9e^{-2}$	$1.9e^{-2}$	99	$1.5e^1$	197	$2.4e^1$
	AEO	$5.6e^{-4}$	$1.9e^{-1}$	$3.4e^{-1}$	98	5.7	197	$2.4e^1$
	MPSO	$4.2e^{-2}$	$2.1e^{-1}$	$1.5e^{-1}$	99	$1.4e^1$	197	$2.4e^1$
	CHSO	$2.8e^1$	$7.4e^1$	$1.7e^1$	99	$4.3e^1$	197	$2.4e^1$
f_2	1D-SOA	$1.7e^{-14}$	$6.7e^{-14}$	$3.0e^{-13}$				
	FWA	$8.7e^{-56}$	$1.7e^{-49}$	$6.1e^{-49}$	99	2.3	99	$2.5e^1$
	ZOA	$2.4e^{-22}$	$2.0e^{-7}$	$2.0e^{-6}$	99	1.0	99	$2.5e^1$
	AEO	$2.3e^{-21}$	$1.2e^{-15}$	$4.2e^{-15}$	99	2.2	99	$2.5e^1$
	MPSO	$7.9e^{-19}$	$1.1e^{-16}$	$5.5e^{-16}$	99	2.3	99	$2.5e^1$
	CHSO	$8.1e^{-277}$	$7.6e^{-3}$	$1.3e^{-2}$	99	6.0	99	$2.5e^1$
f_3	1D-SOA	$6.9e^{-17}$	$4.0e^{-14}$	$7.4e^{-14}$				
	FWA	$3.1e^{-9}$	$2.7e^{-5}$	$4.4e^{-5}$	99	6.1	168	$1.7e^1$
	ZOA	$1.1e^{-3}$	$2.2e^{-2}$	$2.0e^{-2}$	98	$1.1e^1$	168	$1.7e^1$
	AEO	$5.6e^{-8}$	$4.2e^{-2}$	$6.8e^{-2}$	99	6.1	168	$1.7e^1$
	MPSO	$1.2e^{-2}$	2.1	7.0	98	3.0	168	$1.7e^1$
	CHSO	$8.5e^{-4}$	1.1	1.9	99	5.8	168	$1.7e^1$
f_4	1D-SOA	$4.2e^{-17}$	$8.3e^{-3}$	$3.4e^{-2}$				
	FWA	$1.7e^1$	$3.0e^5$	$6.3e^5$	99	4.8	170	$1.7e^1$
	ZOA	$3.8e^6$	$4.9e^7$	$4.0e^7$	99	$1.2e^1$	170	$1.7e^1$
	AEO	$4.6e^5$	$1.7e^8$	$4.4e^8$	99	3.8	170	$1.7e^1$
	MPSO	$2.3e^7$	$2.6e^8$	$3.1e^8$	99	8.3	170	$1.7e^1$
	CHSO	$5.1e^9$	$2.2e^{10}$	$7.1e^9$	99	$3.1e^1$	170	$1.7e^1$
f_5	1D-SOA	$3.7e^{-10}$	$4.1e^{-3}$	$9.3e^{-3}$				
	FWA	$3.1e^{-13}$	$2.8e^{-7}$	$1.2e^{-6}$	99	4.5	144	8.6
	ZOA	$4.0e^{-3}$	$1.2e^{-1}$	$1.4e^{-1}$	99	8.1	144	8.6
	AEO	$2.7e^{-5}$	$4.1e^{-1}$	$3.9e^{-1}$	99	$1.0e^1$	144	8.6
	MPSO	$4.9e^{-2}$	$6.2e^{-1}$	$3.9e^{-1}$	99	$1.6e^1$	144	8.6
	CHSO	9.6	$8.4e^1$	$3.8e^1$	99	$2.2e^1$	144	8.6
f_6	1D-SOA	2.0	2.0	0.0				
	FWA	$1.3e^{10}$	$3.8e^{20}$	$2.4e^{21}$	99	1.6	99	$3.6e^1$
	ZOA	$1.6e^{24}$	$1.1e^{32}$	$6.9e^{32}$	99	1.6	99	$3.6e^1$
	AEO	0.0	$1.3e^5$	$4.9e^5$	98	2.6	99	$3.6e^1$
	MPSO	$6.6e^{-23}$	$2.5e^{31}$	$1.1e^{32}$	99	2.2	99	$3.6e^1$
	CHSO	$3.2e^{16}$	$8.3e^{28}$	$5.3e^{29}$	98	1.6	99	$3.6e^1$
f_7	1D-SOA	$8.4e^{-15}$	$2.2e^{-6}$	$1.6e^{-5}$				
	FWA	$8.5e^{11}$	$4.5e^{18}$	$2.7e^{19}$	99	1.7	98	$2.5e^1$
	ZOA	$1.5e^{24}$	$1.5e^{31}$	$6.0e^{31}$	99	2.6	98	$2.5e^1$
	AEO	0.0	$5.5e^5$	$4.3e^6$	99	1.3	98	$2.5e^1$
	MPSO	$7.4e^{22}$	$6.9e^{30}$	$3.3e^{31}$	98	2.1	98	$2.5e^1$
	CHSO	$1.3e^{17}$	$4.3e^{28}$	$4.0e^{29}$	99	1.1	98	$2.5e^1$
f_8	1D-SOA	$1.6e^{-34}$	$4.8e^{-29}$	$2.9e^{-28}$				
	FWA	$2.5e^{-18}$	$7.0e^{-10}$	$3.0e^{-9}$	99	2.3	141	$1.1e^1$
	ZOA	$5.7e^{-8}$	$1.0e^{-5}$	$1.6e^{-5}$	99	6.2	141	$1.1e^1$
	AEO	$1.3e^{-12}$	$1.1e^{-3}$	$8.4e^{-3}$	99	1.3	141	$1.1e^1$
	MPSO	$2.9e^{-7}$	$3.8e^{-3}$	$1.7e^{-2}$	99	2.2	141	$1.1e^1$
	CHSO	$1.4e^{-13}$	1.1	9.9	99	1.1	141	$1.1e^1$
f_9	1D-SOA	$2.8e^{-14}$	$7.3e^{-1}$	$8.1e^{-1}$				
	FWA	$6.8e^{-13}$	$1.1e^{-7}$	$4.5e^{-7}$	99	9.0	144	3.4
	ZOA	$4.4e^{-2}$	6.9	$1.8e^1$	99	3.5	144	3.4
	AEO	$3.5e^{-5}$	$1.6e^1$	$4.1e^1$	99	3.6	144	3.4
	MPSO	$4.1e^{-2}$	3.4	9.5	100	2.8	144	3.4
	CHSO	$2.8e^1$	$1.5e^2$	$4.3e^1$	99	$3.4e^1$	144	3.4

Table A5. Experimental results of various algorithms and benchmark functions for 30D.

		Best	Mean	Std	<i>d</i>	<i>t</i>	<i>d_{np}</i>	<i>t_{np}</i>
<i>f</i> ₁₀	1D-SOA	3.4e⁻⁴	8.3	7.8				
	FWA	2.6e ¹	2.7e ¹	5.1e ⁻¹	99	2.4e ¹	132	7.6
	ZOA	2.8e ¹	2.9e ¹	9.6e ⁻²	99	2.6e ¹	132	7.6
	AEO	2.5e ¹	2.7e ¹	4.1e ⁻¹	99	2.4e ¹	132	7.6
	MPSO	2.8e ¹	2.9e ¹	2.2e ⁻¹	99	2.6e ¹	132	7.6
	CHSO	2.8e ¹	2.9e ¹	2.8e ⁻¹	99	2.6e ¹	132	7.6
<i>f</i> ₁₁	1D-SOA	3.4e⁻⁴	8.3	7.8	0.0			
	FWA	2.6e ¹	2.7e ¹	5.1e ⁻¹	1.0e ²	2.4e ¹	1.3e ²	7.6
	ZOA	2.8e ¹	2.9e ¹	9.6e ⁻²	9.9e ¹	2.6e ¹	1.3e ²	7.6
	AEO	2.5e ¹	2.7e ¹	4.1e ⁻¹	1.0e ²	2.4e ¹	1.3e ²	7.6
	MPSO	2.8e ¹	2.9e ¹	2.2e ⁻¹	9.9e ¹	2.6e ¹	1.3e ²	7.6
	CHSO	2.8e ¹	2.9e ¹	2.8e ⁻¹	9.9e ¹	2.6e ¹	1.3e ²	7.6
<i>f</i> ₁₂	1D-SOA	2.9	4.0	5.9e ⁻¹				
	FWA	3.9e ⁻⁵¹	2.9e ⁻²	4.5e ⁻²	100	6.8e ¹	98	8.3
	ZOA	1.0e ⁻¹	1.0e ⁻¹	1.9e ⁻⁹	99	6.7e ¹	98	8.3
	AEO	2.1e⁻²⁰	1.2e⁻⁴	1.1e ⁻³	99	6.8e ¹	98	8.3
	MPSO	1.0e ⁻¹	1.6e ⁻¹	6.5e ⁻²	101	6.5e ¹	98	8.3
	CHSO	1.5e ⁻¹	9.7e ⁻¹	4.0e ⁻¹	174	4.3e ¹	98	8.3
<i>f</i> ₁₃	1D-SOA	1.4e⁻⁶	1.1e⁻⁵	1.2e ⁻⁵				
	FWA	5.6e ⁻⁴	5.9e ⁻²	5.3e ⁻²	99	1.1e ¹	190	2.1e ¹
	ZOA	2.7	9.3	4.5	99	2.1e ¹	190	2.1e ¹
	AEO	7.6e ⁻³	2.1e ¹	2.3e ¹	99	9.2	190	2.1e ¹
	MPSO	7.3	1.9e ¹	5.5	99	3.4e ¹	190	2.1e ¹
	CHSO	1.8e ²	3.7e ²	7.4e ¹	99	5.0e ¹	190	2.1e ¹
<i>f</i> ₁₄	1D-SOA	5.2e ⁻¹	1.1	3.1e ⁻¹				
	FWA	1.2e⁻⁴⁸	1.0e⁻³⁹	3.9e ⁻³⁹	99	3.6e ¹	99	1.7e ¹
	ZOA	7.2e ⁻¹⁹	1.4e ⁻¹⁶	2.3e ⁻¹⁶	99	3.6e ¹	99	1.7e ¹
	AEO	4.9e ⁻¹⁹	3.9e ⁻¹⁴	1.8e ⁻¹³	99	3.6e ¹	99	1.7e ¹
	MPSO	8.9e ⁻¹⁷	5.0e ⁻¹⁴	1.7e ⁻¹³	99	3.6e ¹	99	1.7e ¹
	CHSO	1.1e ¹	4.9e ¹	1.7e ¹	99	2.8e ¹	99	1.7e ¹
<i>f</i> ₁₅	1D-SOA	1.0e⁻¹⁵	1.8e⁻¹²	1.1e ⁻¹¹				
	FWA	6.5e ⁻⁹	1.0e ⁻⁶	1.9e ⁻⁶	99	5.3	197	2.4e ¹
	ZOA	1.4e ⁻²	8.5e ⁻²	6.6e ⁻²	98	1.3e ¹	197	2.4e ¹
	AEO	2.6e ⁻³	1.2	2.7	99	4.6	197	2.4e ¹
	MPSO	8.7e ⁻²	4.2e ⁻¹	2.5e ⁻¹	98	1.7e ¹	197	2.4e ¹
	CHSO	3.1e ¹	1.6e ²	7.0e ¹	99	2.3e ¹	197	2.4e ¹
<i>f</i> ₁₆	1D-SOA	3.9e⁻¹⁵⁶	2.4e⁻¹⁴¹	1.8e ⁻¹⁴⁰				
	FWA	4.0e ⁻¹⁴⁶	1.7e ⁻⁸⁵	1.7e ⁻⁸⁴		1.0	197	2.4e ¹
	ZOA	9.0e ⁻⁵⁴	1.7e ⁻⁴⁰	9.4e ⁻⁴⁰	99	1.8	197	2.4e ¹
	AEO	3.6e ⁻⁷⁸	6.5e ⁻³⁰	6.2e ⁻²⁹	99	1.1	197	2.4e ¹
	MPSO	7.1e ⁻⁴⁴	2.9e ⁻²⁵	2.1e ⁻²⁴	99	1.3	197	2.4e ¹
	CHSO	2.0e ⁵	2.4e ⁸	2.9e ⁸	99	8.2	197	2.4e ¹
<i>f</i> ₁₇	1D-SOA	3.6e ³	8.9e ³	1.3e ³				
	FWA	8.9e⁻⁵	1.1	3.9	99	7.0e ¹	99	1.6e ¹
	ZOA	1.6e ¹	3.1e ²	2.2e ²	105	6.7e ¹	99	1.6e ¹
	AEO	1.2e ⁻³	7.4e ¹	2.2e ²	104	6.9e ¹	99	1.6e ¹
	MPSO	1.5e ²	2.3e ⁴	1.1e ⁴	101	1.2e ¹	99	1.6e ¹
	CHSO	1.3e ⁴	2.6e ⁴	6.4e ³	106	2.7e ¹	99	1.6e ¹
<i>f</i> ₁₈	1D-SOA	1.1e ⁻²	2.3	4.8e ⁻¹				
	FWA	6.1e⁻⁴	2.3e⁻³	1.8e ⁻³	99	4.9e ¹	120	4.8
	ZOA	3.5	3.5	9.4e ⁻³	99	2.4e ¹	120	4.8
	AEO	1.7e ⁻¹	1.7	1.1	134	4.8	120	4.8
	MPSO	3.5	3.5	1.8e ⁻³	99	2.5e ¹	120	4.8
	CHSO	3.1	3.4	8.0e ⁻²	104	2.3e ¹	120	4.8
<i>f</i> ₁₉	1D-SOA	4.7e⁻¹⁵	5.5e⁻¹⁰	1.9e ⁻⁹				
	FWA	7.7e ⁻⁹	2.5e ⁻⁵	9.0e ⁻⁵	99	2.8	174	1.8e ¹
	ZOA	2.8e ⁻³	2.4e ⁻²	2.2e ⁻²	99	1.1e ¹	174	1.8e ¹
	AEO	9.0e ⁻⁸	1.1e ⁻¹	2.2e ⁻¹	99	4.9	174	1.8e ¹
	MPSO	1.9e ⁻²	1.5e ⁻¹	1.8e ⁻¹	99	8.4	174	1.8e ¹
	CHSO	1.0e ¹	5.1e ¹	1.4e ¹	99	3.5e ¹	174	1.8e ¹

Table A6. Experimental results of various algorithms and benchmark functions for 30D.

		Best	Mean	Std	d	t	d_{np}	t_{np}
f_{20}	1D-SOA	$8.9e^{-15}$	$1.8e^{-12}$	$3.7e^{-12}$				
	FWA	$8.1e^{-7}$	$9.9e^{-4}$	$2.2e^{-3}$	98	4.5	185	$2.0e^1$
	ZOA	$1.4e^{-1}$	1.5	1.5	99	$1.0e^1$	185	$2.0e^1$
	AEO	$1.1e^{-3}$	6.2	$1.2e^1$	99	5.1	185	$2.0e^1$
	MPSO	$5.1e^{-1}$	$1.3e^1$	$2.3e^1$	98	5.7	185	$2.0e^1$
	CHSO	$1.0e^3$	$2.6e^3$	$9.0e^2$	99	$2.9e^1$	185	$2.0e^1$
f_{21}	1D-SOA	$4.1e^{-16}$	$1.1e^{-9}$	$5.1e^{-9}$				
	FWA	$1.1e^{-11}$	$4.7e^{-6}$	$1.1e^{-5}$	99	4.3	139	2.4
	ZOA	$3.0e^{-7}$	$1.0e^{-4}$	$1.7e^{-4}$	99	6.0	139	2.4
	AEO	$1.5e^{-9}$	$5.0e^{-3}$	$2.7e^{-2}$	99	1.9	139	2.4
	MPSO	$4.1e^{-6}$	$2.2e^{-2}$	$4.9e^{-2}$	99	4.4	139	2.4
	CHSO	$2.3e^{-12}$	$3.0e^{-2}$	$1.6e^{-1}$	99	1.9	139	2.4
f_{22}	1D-SOA	$4.0e^{-14}$	1.5	5.9				
	FWA	$6.9e^{-7}$	$8.8e^{-2}$	$1.9e^{-1}$	99	2.3	169	3.1
	ZOA	6.0	$4.9e^1$	$3.6e^1$	104	$1.3e^1$	169	3.1
	AEO	$1.3e^{-2}$	$4.5e^2$	$9.3e^2$	99	4.9	169	3.1
	MPSO	$2.7e^1$	$1.4e^3$	$3.6e^3$	99	3.9	169	3.1
	CHSO	$6.9e^3$	$3.2e^4$	$1.0e^4$	99	$3.1e^1$	169	3.1
f_{23}	1D-SOA	0.0	0.0	0.0				
	FWA	0.0	0.0	0.0				
	ZOA	0.0	0.0	0.0				
	AEO	0.0	0.0	0.0				
	MPSO	0.0	0.0	0.0				
	CHSO	0.0	0.0	0.0				
f_{24}	1D-SOA	$6.8e^{-9}$	$2.2e^{-2}$	$1.2e^{-2}$				
	FWA	$3.7e^{-7}$	$2.3e^{-4}$	$3.2e^{-4}$	99	$1.8e^1$	152	$1.3e^1$
	ZOA	$1.4e^{-1}$	$5.0e^{-1}$	$1.2e^{-1}$	100	$4.0e^1$	152	$1.3e^1$
	AEO	$3.1e^{-4}$	$5.7e^{-1}$	$2.6e^{-1}$	99	$2.1e^1$	152	$1.3e^1$
	MPSO	$1.5e^{-1}$	$7.6e^{-1}$	$1.0e^{-1}$	101	$7.1e^1$	152	$1.3e^1$
	CHSO	$2.7e^{-1}$	$3.9e^{-1}$	$6.4e^{-2}$	105	$5.7e^1$	152	$1.3e^1$
f_{25}	1D-SOA	$2.8e^{-13}$	$4.8e^{-8}$	$1.5e^{-7}$				
	FWA	$6.1e^{-12}$	$1.5e^{-6}$	$6.8e^{-6}$	99	2.2	195	6.4
	ZOA	$9.8e^{-7}$	$4.9e^{-4}$	$6.6e^{-4}$	99	7.4	195	6.4
	AEO	$2.2e^{-5}$	$1.8e^{-2}$	$5.3e^{-2}$	99	3.4	195	6.4
	MPSO	$1.0e^{-3}$	$4.7e^3$	$4.5e^4$	99	1.0	195	6.4
	CHSO	$6.3e^5$	$4.2e^{12}$	$2.0e^{13}$	99	2.1	195	6.4
f_{26}	1D-SOA	$3.5e^{-12}$	$3.7e^{-12}$	$2.0e^{-13}$				
	FWA	$6.5e^{-9}$	$7.8e^{-8}$	$6.4e^{-8}$	99	$1.2e^1$	99	$1.7e^1$
	ZOA	$9.5e^{-8}$	$2.5e^{-6}$	$2.8e^{-6}$	99	9.0	99	$1.7e^1$
	AEO	$3.5e^{-12}$	$2.0e^{-11}$	$1.1e^{-11}$	99	$1.6e^1$	99	$1.7e^1$
	MPSO	$1.1e^{-6}$	$2.0e^{-4}$	$2.9e^{-4}$	99	7.0	99	$1.7e^1$
	CHSO	0.0	$4.7e^{-9}$	$3.5e^{-8}$	99	1.4	99	$1.7e^1$
f_{27}	1D-SOA	1.0	1.0	0.0				
	FWA	1.0	1.0	$3.6e^{-12}$	$9.9e^1$	$3.3e^1$	0.0	$9.9e^1$
	ZOA	$3.8e^{-11}$	$1.8e^{-10}$	$1.1e^{-10}$	$9.9e^1$	$9.1e^{10}$	$9.1e^9$	$9.9e^1$
	AEO	$-9.1e^{-1}$	$-2.8e^{-2}$	$1.3e^{-1}$	$9.9e^1$	$7.6e^1$	7.9	$9.9e^1$
	MPSO	1.0	1.0	$3.4e^{-9}$	$9.9e^1$	$1.3e^1$	0.0	$9.9e^1$
	CHSO	1.0	1.0	$5.7e^{-10}$	$9.9e^1$	$1.0e^1$	0.0	$9.9e^1$
f_{28}	1D-SOA	$1.9e^2$	$3.7e^2$	$5.5e^1$				
	FWA	$1.7e^{-4}$	$8.9e^{-1}$	1.5	99	$6.8e^1$	99	$1.6e^1$
	ZOA	$9.8e^{-1}$	$1.5e^1$	7.9	103	$6.5e^1$	99	$1.6e^1$
	AEO	$2.7e^{-3}$	$1.0e^1$	$4.2e^1$	185	$5.3e^1$	99	$1.6e^1$
	MPSO	$6.5e^{-1}$	$1.9e^4$	$1.1e^5$	99	1.6	99	$1.6e^1$
	CHSO	$2.0e^2$	$1.1e^3$	$3.0e^3$	99	2.3	99	$1.6e^1$
f_{29}	1D-SOA	$1.9e^2$	$3.7e^2$	$5.5e^1$				
	FWA	$1.7e^{-4}$	$8.9e^{-1}$	1.5	$9.9e^1$	$6.8e^1$	$9.9e^1$	$1.6e^1$
	ZOA	$9.8e^{-1}$	$1.5e^1$	7.9	$1.0e^2$	$6.5e^1$	$9.9e^1$	$1.6e^1$
	AEO	$2.7e^{-3}$	$1.0e^1$	$4.2e^1$	$1.9e^2$	$5.3e^1$	$9.9e^1$	$1.6e^1$
	MPSO	$6.5e^{-1}$	$1.9e^4$	$1.1e^5$	$9.9e^1$	1.6	$9.9e^1$	$1.6e^1$
	CHSO	$2.0e^2$	$1.1e^3$	$3.0e^3$	$9.9e^1$	2.3	$9.9e^1$	$1.6e^1$

Table A7. Experimental results of various algorithms and benchmark functions for 2D, for functions with non-zero optimum.

		Best	Mean	Std	<i>d</i>	<i>t</i>	<i>d_{np}</i>	<i>t_{np}</i>
<i>f</i> ₃₀	1D-SOA	−1.0	−1.0	1.3e ^{−3}				
	FWA	−1.0	−1.0	1.1e ^{−13}	99	1.5	158	4.1
	ZOA	−1.0	−1.0	4.8e ^{−9}	99	1.5	158	4.1
	AEO	−1.0	−1.0	3.1e ^{−6}	99	1.5	158	4.1
	MPSO	−1.0	−1.0	2.2e ^{−5}	99	1.4	158	4.1
<i>f</i> ₃₁	1D-SOA	3.0	8.6	1.0e ¹				
	FWA	3.0	4.6	6.4	168	3.4	176	9.1
	ZOA	3.0	1.5e ¹	1.9e ¹	149	3.0	176	9.1
	AEO	3.0	6.4	1.3e ¹	187	1.4	176	9.1
	MPSO	3.1	1.3e ¹	1.4e ¹	180	2.7	176	9.1
<i>f</i> ₃₂	1D-SOA	1.1	2.4e ³	5.9e ³				
	FWA	1.0	1.4	1.6	99	4.1	185	6.8
	ZOA	1.0	4.6e ¹	1.3e ²	99	4.0	185	6.8
	AEO	1.5	2.5e ²	5.7e ²	100	3.7	185	6.8
	MPSO	1.2	6.6e ²	1.9e ³	119	2.8	185	6.8
<i>f</i> ₃₃	1D-SOA	−1.0	−9.4e ^{−1}	2.4e ^{−1}				
	FWA	−1.0	−1.0	4.8e ^{−6}	99	2.5	98	2.2e ¹
	ZOA	−1.0	−9.5e ^{−1}	2.2e ^{−1}	196	3.1e ^{−1}	98	2.2e ¹
	AEO	−1.0	−1.0	4.5e ^{−14}	99	2.5	98	2.2e ¹
	MPSO	−1.0	− 9.3e^{−1}	6.7e ^{−2}	114	5.8e ^{−1}	98	2.2e ¹
<i>f</i> ₃₄	1D-SOA	1.244e ²	1.387e ²	1.671e ¹				
	FWA	1.244e ²	1.262e²	2.352	102	7	191	5
	ZOA	1.244e ²	1.463e ²	4.155e ¹	130	1	191	5
	AEO	1.244e ²	1.417e ²	2.493e ¹	173	1	191	5
	MPSO	1.250e ²	1.845e ²	4.020e ¹	132	10	191	5
<i>f</i> ₃₅	1D-SOA	9.0e ^{−1}	1.0	2.5e ^{−2}				
	FWA	9.0e ^{−1}	9.0e^{−1}	2.3e ^{−3}	100	3.8e ¹	193	8.9
	ZOA	9.0e ^{−1}	9.6e ^{−1}	4.9e ^{−2}	148	6.7	193	8.9
	AEO	9.0e ^{−1}	1.0	4.1e ^{−2}	165	3.7e ^{−1}	193	8.9
	MPSO	9.0e ^{−1}	1.0	5.6e ^{−2}	138	4.1	193	8.9
<i>f</i> ₃₆	1D-SOA	− 3456	−2884					
	FWA	−3455	− 3450	19	99	5	165	0
	ZOA	−3455	−2234	1452	184	3	165	0
	AEO	−3455	−1520	1555	178	7	165	0
	MPSO	−3452	−725	3447	119	5	165	0
<i>f</i> ₃₇	1D-SOA	−7.833e ¹	−7.830e ¹	2.513e ^{−1}				
	FWA	−7.833e ¹	− 7.832e¹	2.713e ^{−2}	101		110	7
	ZOA	−7.833e ¹	−7.201e ¹	7.658	99	8	110	7
	AEO	−7.833e ¹	−7.657e ¹	4.304	99	4	110	7
	MPSO	−7.820e ¹	−7.256e ¹	5.408	99	10	110	7

Appendix C. Benchmark Functions

Table A8. Benchmark functions.

Function	Search Space, Ω	Optimal Solution, \vec{x}^*	Optimum, $f(\vec{x}^*)$	C_{neigh}	C_{neigh_l}
$f_0(\vec{x})$: Ackley [56–58]	[−32.768, 32.768]	$[0, 0]^{D^*}$	0	0.5	0.01
		$f_0(\vec{x}) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_i^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_i^D \cos(2\pi x_i)\right) + 20 + \exp(1)$			
$f_1(\vec{x})$: Ackley 2 [58]	[−32, 32] ^D	$[0, 0]^D$	0	5	0.1
		$f_1(\vec{x}) = 200 \left(1 - \exp\left(-0.02 \sqrt{\sum_i^D x_i^2}\right)\right)$			
$f_2(\vec{x})$: Alpine 1 [58]	[−10, 10] ^D	$[0, 0]^D$	0	0.5	0.1
		$f_2(\vec{x}) = \sum_i^n x_i \sin(x_i) + 0.1x_i $			

Table A8. Cont.

Function	Search Space, Ω	Optimal Solution, \vec{x}^*	Optimum, $f(\vec{x}^*)$	C_{neigh}	C_{neigh_l}
$f_3(\vec{x})$: Brown [58]	$[-1, 4]^D$	$[0, 0]^D$ $f_3(\vec{x}) = \sum_{i=1}^{D-1} \left((x_i^2)^{(x_{i+1}^2+1)} + (x_{i+1}^2)^{(x_i^2+1)} \right)$	0	0.5	0.1
$f_4(\vec{x})$: Cigar [20]	$[-100, 100]^D$	$[0, 0]^D$ $f_4(\vec{x}) = x_1^2 + 10^6 \sum_{i=2}^D x_i^2$	0	0.5	0.0001
$f_5(\vec{x})$: Ellipse [20]	$[-100, 100]^D$	$[0, 0]^D$ $f_5(\vec{x}) = \sum_i^D 10^4 \frac{i-1}{b-1} x_i^2$	0	0.5	0.01
$f_6(\vec{x})$: Griewank [20]	$[-600, 600]^D$	$[0, 0]^D$ $f_6(\vec{x}) = \frac{1}{4000} \sum_i^D x_i^2 - \prod_i^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	0	250	15
$f_7(\vec{x})$: Mishra 1 [59]	$[-1, 1]^D$	$[1, 1]^D$ $f_7(\vec{x}) = \left(1 + D - \sum_i^{D-1} x_i\right)^{D - \sum_i^{D-1} x_i} - 2$	0	0.5	0.01
$f_8(\vec{x})$: Mishra 2 [59]	$[-1, 1]^D$	$[1, 1]^D$ $f_8(\vec{x}) = (1 + gn)^{gn} - 2, gn = D - \sum_i^{D-1} \left(\frac{x_i + x_{i+1}}{2}\right)$	0	0.5	0.01
$f_9(\vec{x})$: Quartic [58]	$[-1.28, 1.28]^D$	$[0, 0]^D$ $f_9(\vec{x}) = \sum_i^D ix_i^4$	0	1.28	0.2
$f_{10}(\vec{x})$: Rastrigin [56,57]	$[-5.12, 5.12]^D$	$[0, 0]^D$ $f_{10}(\vec{x}) = 10D + \sum_i^D (x_i^2 - 10\cos(2\pi x_i))$	0	5	2
$f_{11}(\vec{x})$: Rosenbrock [23]	$[-2.048, 2.048]^D$	$[1, 1]^D$ $f_{11}(\vec{x}) = \sum_i^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	0	3	0.5

* The super-index D refers to a D -dimensional vector.

Table A9. Benchmark functions.

Function	Search Space, Ω	Optimal Solution, \vec{x}^*	Optimum, $f(\vec{x}^*)$	C_{neigh}	C_{neigh_l}
$f_{12}(\vec{x})$: Salomon [58]	$[-100, 100]^D$	$[0, 0]^D$ $f_{12}(\vec{x}) = 1 - \cos\left(2\pi\sqrt{\sum_i^D x_i^2}\right) + 0.1\sqrt{\sum_i^D x_i^2}$	0	0.01	0.001
$f_{13}(\vec{x})$: Schwefel [23]	$[-500, 500]^D$	$[420.9687, 420.9687]^D$ $f_{13}(\vec{x}) = 418.9829D - \sum_i^D x_i \sin\left(\sqrt{ x_i }\right)$	0	500	250
$f_{14}(\vec{x})$: Schwefel 2.20 [58]	$[-100, 100]^D$	$[0, 0]^D$ $f_{14}(\vec{x}) = \sum_i^D x_i $	0	5	2
$f_{15}(\vec{x})$: Schwefel 2.21 [23,58]	$[-100, 100]^D$	$[0, 0]^D$ $f_{15}(\vec{x}) = \max(x_i)$	0	0.5	0.01
$f_{16}(\vec{x})$: Schwefel 2.22 [58]	$[-100, 100]^D$	$[0, 0]^D$ $f_{16}(\vec{x}) = \sum_i^D x_i + \prod_i^D x_i $	0	50	2
$f_{17}(\vec{x})$: Schwefel 2.23 [58]	$[-10, 10]^D$	$[0, 0]^D$ $f_{17}(\vec{x}) = \sum_i^D x_i^{10}$	0	10	5
$f_{18}(\vec{x})$: Schwefel Double Sum [23,58]	$[-65.536, 65.536]^D$	$[0, 0]^D$	0	5	1
$f_{19}(\vec{x})$: Sinusoidal [59]	$[0, 180]^D$	$[0, 0]^D$ $f_{19}(\vec{x}) = A + 1 - \left\{A \prod_i^D \sin(x_i - Z) + \prod_i^D \sin[B(x_i - Z)]\right\}, A = 2.5, B = 5, Z = 30$	0	5	0.1
$f_{20}(\vec{x})$: Sphere [56–58]	$[-5.12, 5.12]^D$	$[0, 0]^D$ $f_{20}(\vec{x}) = \sum_i^D x_i^2$	0	0.5	0.01
$f_{21}(\vec{x})$: Sum Squares [58]	$[-10, 10]^D$	$[0, 0]^D$ $f_{21}(\vec{x}) = \sum_i^D ix_i^2$	0	10	5
$f_{22}(\vec{x})$: Sum of Different Powers [29]	$[-1, 1]^D$	$[0, 0]^D$ $f_{22}(\vec{x}) = x_i ^{i+1}$	0	0.5	0.01
$f_{23}(\vec{x})$: Step [23]	$[-5, 5]^D$	$[0, 0]^D$ $f_{23}(\vec{x}) = \sum_i^D \text{floor}(x_i + 0.5)^2$	0	5	1
$f_{24}(\vec{x})$: Tablet [20]	$[-100, 100]^D$	$[0, 0]^D$ $f_{24}(\vec{x}) = 10^4 x_1^2 + \sum_{i=2}^D x_i^2$	0	0.5	0.001

* The super-index D refers to a D -dimensional vector.

Table A10. Benchmark functions.

Functions	Search Space, Ω	Optimal Solution, \vec{x}^*	Optimum, $f(\vec{x}^*)$		
$f_{25}(\vec{x})$: Wavy [58]	$[-\pi, \pi]^D$	$[0, 0]^D$	0	$\pi/2$	1
$f_{26}(\vec{x})$: XinShe Yang [60]	$[-5, 5]^D$	$f_{25}(\vec{x}) = 1 - \frac{1}{D} \sum_i^D \cos(10x_i) \exp(x_i^2/2)$ $[0, 0]^D$	0	0.5	0.001
$f_{27}(\vec{x})$: XinShe Yang N.2 [60]	$[-2\pi, 2\pi]^D$	$f_{26}(\vec{x}) = x_i ^i$ $[0, 0]^D$	0	2π	π
$f_{28}(\vec{x})$: XinShe Yang N.4 [60]	$[-10, 10]^D$	$f_{27}(\vec{x}) = \left(\sum_i^D x_i \right) \exp\left(-\sum_i^D \sin(x_i^2)\right)$ $[0, 0]^D$	0	0.5	0.2
$f_{29}(\vec{x})$: Zakharov [58]	$[-5, 10]^D$	$f_{28}(\vec{x}) = \left(\sum_i^D \sin^2(x_i) - \exp(-\sum_i^D x_i^2)\right) \exp\left(-\sum_i^D \sin^2 \sqrt{ x_i }\right) + 1$ $[0, 0]^D$	0	1	0.001
		$f_{29}(\vec{x}) = \sum_i^D x_i^2 + \left(\sum_i^D 0.5ix_i\right)^2 + \left(\sum_i^D 0.5ix_i\right)^4$			

Table A11. Non-zero benchmark functions.

Function	Search Space, Ω	Optimal Solution, \vec{x}^*	Optimum, $f(\vec{x}^*)$	C_{neigh}	C_{neigh_l}
D-dimensional functions					
$f_7(\vec{x})$: Mishra 1 [59]	$[-1, 1]^D$	$[1, 1]^D$	2	0.5	0.01
$f_8(\vec{x})$: Mishra 2 [59]	$[-1, 1]^D$	$f_7(\vec{x}) = \left(1 + D - \sum_i^{D-1} x_i\right)^{D - \sum_i^{D-1} x_i}$ $[1, 1]^D$	2	0.5	0.01
		$f_8(\vec{x}) = (1 + gn)^{gn}, gn = D - \sum_i^{D-1} \left(\frac{x_i + x_{i+1}}{2}\right)$			
2-dimensional functions					
$f_{30}(\vec{x})$: Exponential [58]	$[-1, 1]^D$	$[0, 0]$	1	0.05	0.001
$f_{31}(\vec{x})$: Goldstein Price [32]	$[-2, 2]^D$	$f_{30}(\vec{x}) = \left(1 + D - \sum_i^{D-1} x_i\right)^{D - \sum_i^{D-1} x_i}$ $[0, -1]^D$	3	0.06	0.01
$f_{32}(\vec{x})$: Bartels Conn [58]	$[-500, 500]^D$	$f_{31}(\vec{x}) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times$ $[0, 0]$ $\times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	1	10	0.01
$f_{33}(\vec{x})$: Easom [58]	$[-100, 100]^D$	$f_{32}(\vec{x}) = x_1^2 + x_2^2 + x_1x_2 $ $[0, -1]^D$	3	70	40
$f_{34}(\vec{x})$: Jennrich-Sampson [58]	$[-1, 1]^D$	$f_{33}(\vec{x}) = -\cos(x_1)\cos(x_2)\exp[-(x_1 - \pi)^2 - (x_2 - \pi)^2]$ $[0.257825]^D$	124.3612	0.05	0.001
$f_{35}(\vec{x})$: Price 2 [58]	$[-10, 10]^D$	$f_{34}(\vec{x}) = \sum_{i=1}^{10} (2 + 2i - (e^{ix_1} + e^{ix_2})^2)$ $[0]^D$	0.9	0.05	0.0001
$f_{36}(\vec{x})$: Schwefel 2.36 [58]	$[0, 500]^D$	$f_{35}(\vec{x}) = 1 + \sin^2(x_1) + \sin^2(x_2) - 0.1\exp^{-x_1^2 - x_2^2}$ $[12]^D$	-3456	5	0.01
$f_{37}(\vec{x})$: Styblinski-Tang [58]	$[-5, 5]^D$	$f_{36}(\vec{x}) = -x_1x_2(72 - 2x_1 - 2x_2)$ $[-2.903534]^D$	-78.332	5	0.01
		$f_{37}(\vec{x}) = \frac{1}{2} \sum_{i=1}^2 (x_i^4 - 16x_i^2 + 5x_i)$			

* The super-index D refers to a D -dimensional vector.

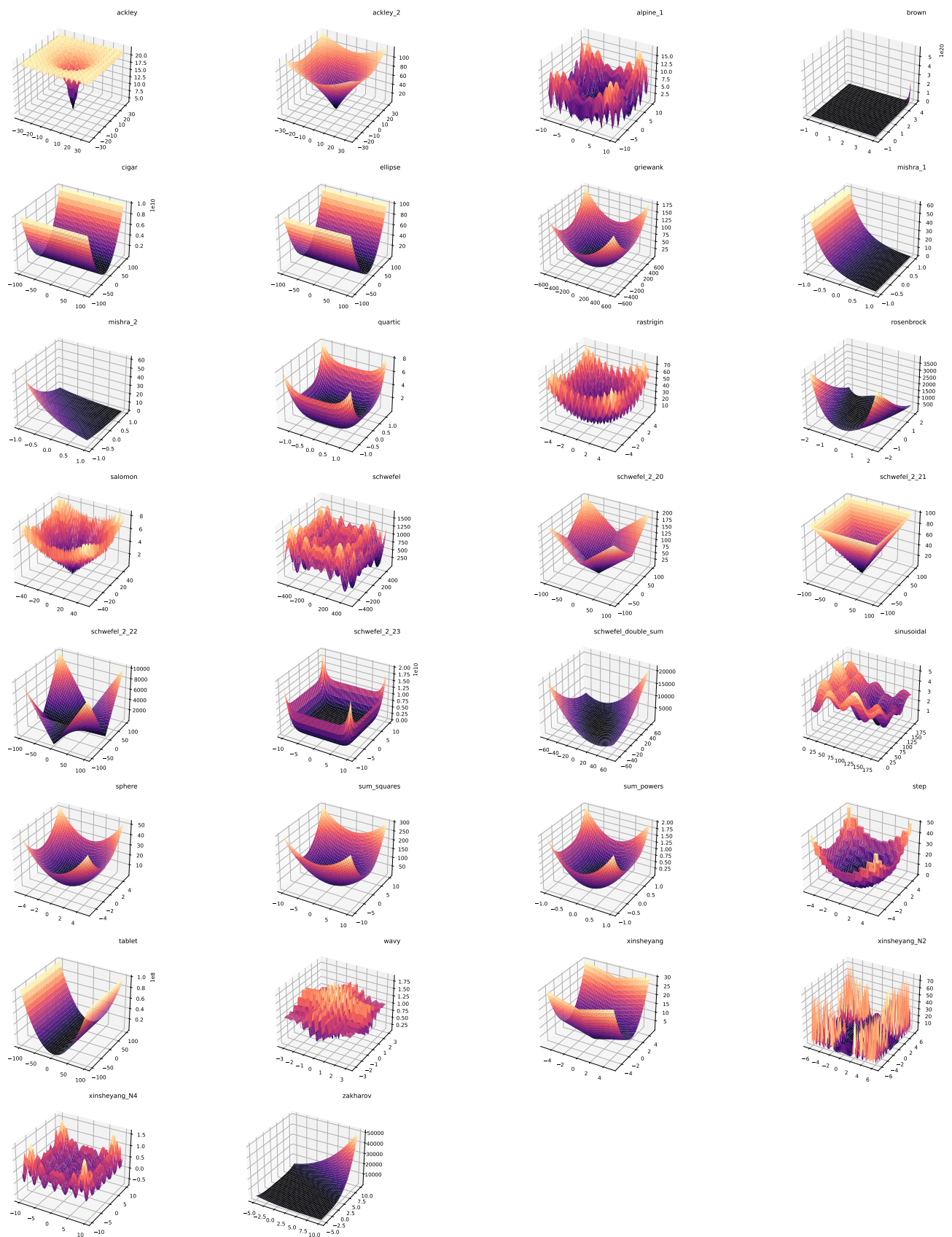


Figure A4. Benchmark functions studied in this paper.

References

- Li, G.; Shuang, F.; Zhao, P.; Le, C. An improved butterfly optimization algorithm for engineering design problems using the cross-entropy method. *Symmetry* **2019**, *11*, 1049. [\[CrossRef\]](#)
- Talatahari, S.; Azizi, M.; Gandomi, A.H. Material generation algorithm: A novel metaheuristic algorithm for optimization of engineering problems. *Processes* **2021**, *9*, 859. [\[CrossRef\]](#)
- Gendreau, M.; Potvin, J.Y. *Handbook of Metaheuristics*; Springer: Berlin/Heidelberg, Germany, 2010; Volume 2. [\[CrossRef\]](#)
- Cruz-Reyes, L.; Quiroz C, M.; F Alvim, A.C.; Fraire Huacuja, H.J.; Gómez S, C.; Torres-Jiménez, J. Efficient Hybrid Grouping Heuristics for the Bin Packing Problem. *Comput. Syst.* **2012**, *16*, 349–360.
- Navas, M.M.; Urbaneja, A.J.N. Metaheurísticas multiobjetivo adaptativas. *Comput. Syst.* **2013**, *17*, 53–62.
- Beheshti, Z.; Shamsuddin, S.M.H. A review of population-based meta-heuristic algorithms. *Int. J. Adv. Soft Comput. Appl.* **2013**, *5*, 1–35.
- Faridmehr, I.; Nehdi, M.L.; Davoudkhani, I.F.; Poolad, A. Mountaineering Team-Based Optimization: A Novel Human-Based Metaheuristic Algorithm. *Mathematics* **2023**, *11*, 1273. [\[CrossRef\]](#)
- Mohammadi-Balani, A.; Nayeri, M.D.; Azar, A.; Taghizadeh-Yazdi, M. Golden eagle optimizer: A nature-inspired metaheuristic algorithm. *Comput. Ind. Eng.* **2021**, *152*, 107050. [\[CrossRef\]](#)
- Polap, D.; Woźniak, M. Polar bear optimization algorithm: Meta-heuristic with fast population movement and dynamic birth and death mechanism. *Symmetry* **2017**, *9*, 203. [\[CrossRef\]](#)
- Meng, X.; Liu, Y.; Gao, X.; Zhang, H. A new bio-inspired algorithm: Chicken swarm optimization. In Proceedings of the International Conference in Swarm Intelligence, Hefei, China, 17–20 October 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 86–94. [\[CrossRef\]](#)
- Abualigah, L.; Yousri, D.; Abd Elaziz, M.; Ewees, A.A.; Al-Qaness, M.A.; Gandomi, A.H. Aquila optimizer: A novel meta-heuristic optimization algorithm. *Comput. Ind. Eng.* **2021**, *157*, 107250. [\[CrossRef\]](#)
- Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [\[CrossRef\]](#)
- Yang, X.S.; Deb, S. Multiobjective cuckoo search for design optimization. *Comput. Oper. Res.* **2013**, *40*, 1616–1624. [\[CrossRef\]](#)
- James, J.; Li, V.O. A social spider algorithm for global optimization. *Appl. Soft Comput.* **2015**, *30*, 614–627. [\[CrossRef\]](#)
- Yang, X.S.; Slowik, A. Firefly algorithm. In *Swarm Intelligence Algorithms*; CRC Press: Boca Raton, FL, USA, 2020; pp. 163–174.
- Pierezan, J.; Coelho, L.D.S. Coyote optimization algorithm: A new metaheuristic for global optimization problems. In Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8. [\[CrossRef\]](#)
- Chu, S.C.; Tsai, P.W.; Pan, J.S. Cat swarm optimization. In Proceedings of the Pacific Rim International Conference on Artificial Intelligence, Guilin, China, 7–11 August 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 854–858. [\[CrossRef\]](#)
- Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948. [\[CrossRef\]](#)
- Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. GSA: A gravitational search algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [\[CrossRef\]](#)
- Tan, Y.; Zhu, Y. Fireworks Algorithm for Optimization. In Proceedings of the Advances in Swarm Intelligence, First International Conference, ICSI 2010, Beijing, China, 12–15 June 2010; Proceedings, Part I; Tan, Y., Shi, Y., Tan, K.C., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6145, pp. 355–364. [\[CrossRef\]](#)
- Eiben, A.E.; Smith, J.E. *Introduction to Evolutionary Computing*; Springer: Berlin/Heidelberg, Germany, 2003; Volume 53. [\[CrossRef\]](#)
- Eiben, A.E.; Schippers, C.A. *Introduction to Evolutionary Computing. Fundamenta Informaticae*; Elsevier: Amsterdam, The Netherlands, 1998; Volume 35, pp. 1–4.
- Simon, D. *Evolutionary Optimization Algorithms*; John Wiley & Sons: Hoboken, NJ, USA, 2013.
- Garcia-Gonzalo, E.; Fernandez-Martinez, J.L. A brief historical review of particle swarm optimization (PSO). *J. Bioinform. Intell. Control* **2012**, *1*, 3–16. [\[CrossRef\]](#)
- Ramírez-Ochoa, D.D.; Pérez-Domínguez, L.A.; Martínez-Gómez, E.A.; Luviano-Cruz, D. PSO, a swarm intelligence-based evolutionary algorithm as a decision-making strategy: A review. *Symmetry* **2022**, *14*, 455. [\[CrossRef\]](#)
- Storn, R.; Price, K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341. [\[CrossRef\]](#)
- Opara, K.R.; Arabas, J. Differential Evolution: A survey of theoretical analyses. *Swarm Evol. Comput.* **2019**, *44*, 546–558. [\[CrossRef\]](#)
- Ma, Z.; Yuan, X.; Han, S.; Sun, D.; Ma, Y. Improved chaotic particle swarm optimization algorithm with more symmetric distribution for numerical function optimization. *Symmetry* **2019**, *11*, 876. [\[CrossRef\]](#)
- Özcan, H. Comparison of particle swarm and differential evolution optimization algorithms considering various benchmark functions. *Politek. Derg.* **2017**, *20*, 899–905. [\[CrossRef\]](#)
- Deng, W.; Shang, S.; Cai, X.; Zhao, H.; Song, Y.; Xu, J. An improved differential evolution algorithm and its application in optimization problem. *Soft Comput.* **2021**, *25*, 5277–5298. [\[CrossRef\]](#)
- Sun, W.Z.; Wang, J.S.; Wei, X. An improved whale optimization algorithm based on different searching paths and perceptual disturbance. *Symmetry* **2018**, *10*, 210. [\[CrossRef\]](#)
- Zhao, W.; Wang, L.; Zhang, Z. Artificial ecosystem-based optimization: A novel nature-inspired meta-heuristic algorithm. *Neural Comput. Appl.* **2020**, *32*, 9383–9425. [\[CrossRef\]](#)
- Chou, J.S.; Truong, D.N. A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean. *Appl. Math. Comput.* **2021**, *389*, 125535. [\[CrossRef\]](#)

34. Talatahari, S.; Azizi, M. Chaos game optimization: A novel metaheuristic algorithm. *Artif. Intell. Rev.* **2021**, *54*, 917–1004. [\[CrossRef\]](#)
35. Trojovská, E.; Dehghani, M.; Trojovský, P. Zebra Optimization Algorithm: A New Bio-Inspired Optimization Algorithm for Solving Optimization Algorithm. *IEEE Access* **2022**, *10*, 49445–49473. [\[CrossRef\]](#)
36. Braik, M. Chameleon Swarm Algorithm: A Bio-inspired Optimizer for Solving Engineering Design Problems. *Expert Syst. Appl.* **2021**, *174*, 114685. [\[CrossRef\]](#)
37. Dehghani, M.; Trojovsky, P. Serval Optimization Algorithm: A New Bio-Inspired Approach for Solving Optimization Problems. *Biomimetics* **2022**, *7*, 204. [\[CrossRef\]](#)
38. Trojovský, P.D. A new bio-inspired metaheuristic algorithm for solving optimization problems based on walruses behavior. *Sci. Rep.* **2023**, *13*, 8775. [\[CrossRef\]](#)
39. Hadi, A.; Wagdy, A.; Jambi, K. Single-Objective Real-Parameter Optimization: Enhanced LSHADE-SPACMA Algorithm. In *Heuristics for Optimization and Learning*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 103–121. [\[CrossRef\]](#)
40. Martínez-Álvarez, F.; Cortés, G.; Torres, J.; Gutiérrez-Avilés, D.; Melgar-García, L.; Pérez-Chacón, R.; Rubio-Escudero, C.; Riquelme, J.; Troncoso, A. Coronavirus Optimization Algorithm: A Bioinspired Metaheuristic Based on the COVID-19 Propagation Model. *Big Data* **2020**, *8*, 308–322. [\[CrossRef\]](#)
41. Wagdy, A.; Hadi, A.; Agrawal, P.; Sallam, K.; Khater, A. Gaining-Sharing Knowledge Based Algorithm with Adaptive Parameters Hybrid with IMODE Algorithm for Solving CEC 2021 Benchmark Problems. In Proceedings of the 2021 IEEE Congress on Evolutionary Computation (CEC), Kraków, Poland, 28 June–1 July 2021; pp. 841–848. [\[CrossRef\]](#)
42. Shan-Fan, J.; Wu-Xiong, S.; Zhuo-Wang, J.; Long-Gong, C. IMODE: Improving Multi-Objective Differential Evolution Algorithm. In Proceedings of the 2008 Fourth International Conference on Natural Computation, Jinan, China, 18–20 October 2008; Volume 1, pp. 212–216. [\[CrossRef\]](#)
43. Civicioglu, P. Backtracking search optimization algorithm for numerical optimization problems. *Appl. Math. Comput.* **2013**, *219*, 8121–8144. [\[CrossRef\]](#)
44. Coello, C.A.C.; Lamont, G.B.; Van Veldhuizen, D.A. *Evolutionary Algorithms for Solving Multi-Objective Problems*; Springer: Berlin/Heidelberg, Germany, 2007; Volume 5.
45. Luc, D.T.; Luc, D.T. Linear Programming. In *Multiobjective Linear Programming: An Introduction*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 49–82. [\[CrossRef\]](#)
46. Deb, K. Multi-objective optimization. In *Search Methodologies*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 403–449. [\[CrossRef\]](#)
47. Brežočnik, L.; Fister J, I.; Podgorelec, V. Swarm intelligence algorithms for feature selection: A review. *Appl. Sci.* **2018**, *8*, 1521. [\[CrossRef\]](#)
48. Engelbrecht, A.P. *Computational Intelligence: An Introduction*; John Wiley & Sons: Hoboken, NJ, USA, 2007.
49. SISDevelop. SwarmPackagePy a Swarm-Based Optimization Algorithms Package for Python. 2017. Available online: <https://github.com/SISDevelop/SwarmPackagePy/> (accessed on 1 January 2023).
50. Karaboga, D.; Gorkemli, B.; Ozturk, C.; Karaboga, N. A comprehensive survey: Artificial bee colony (ABC) algorithm and applications. *Artif. Intell. Rev.* **2014**, *42*, 21–57. [\[CrossRef\]](#)
51. Yang, X.S.; He, X. Bat algorithm: Literature review and applications. *Int. J. Bio-Inspired Comput.* **2013**, *5*, 141–149. [\[CrossRef\]](#)
52. Passino, K.M. Bacterial foraging optimization. In *Innovations and Developments of Swarm Intelligence Applications*; IGI Global: Hershey, PA, USA, 2012; pp. 219–234. [\[CrossRef\]](#)
53. Deep, K.; Bansal, J. Mean particle swarm optimization for function optimization. *Int. J. Comput. Intell. Stud.* **2009**, *1*, 72–92. [\[CrossRef\]](#)
54. Luke, S. *Essentials of Metaheuristics*, 2nd ed.; Lulu: Morrisville, NC, USA, 2013. Available online: [http://cs.gmu.edu/~sim\\$ean/book/metaheuristics/](http://cs.gmu.edu/~sim$ean/book/metaheuristics/) (accessed on 1 January 2023).
55. Chahar, V.; Chhabra, J.; Kumar, D. Optimal Choice of Parameters for Fireworks Algorithm. *Procedia Comput. Sci.* **2015**, *70*, 334–340. [\[CrossRef\]](#)
56. Liang, J.J.; Qin, A.K.; Suganthan, P.N.; Baskar, S. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans. Evol. Comput.* **2006**, *10*, 281–295. [\[CrossRef\]](#)
57. Nickabadi, A.; Ebadzadeh, M.M.; Safabakhsh, R. A novel particle swarm optimization algorithm with adaptive inertia weight. *Appl. Soft Comput.* **2011**, *11*, 3658–3670. [\[CrossRef\]](#)
58. Jamil, M.; Yang, X.S. A literature survey of benchmark functions for global optimization problems. *Int. J. Math. Model. Numer. Optim.* **2013**, *4*, 150–194. [\[CrossRef\]](#)
59. Gao, Z.M.; Zhao, J.; Hu, Y.R.; Chen, H.F. The challenge for the nature-inspired global optimization algorithms: Non-symmetric benchmark functions. *IEEE Access* **2021**, *9*, 106317–106339. [\[CrossRef\]](#)
60. Al-Roomi, A.R. *Unconstrained Single-Objective Benchmark Functions Repository*; Dalhousie University, Electrical and Computer Engineering: Halifax, NS, Canada, 2015.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.