

Article

Light Spectrum Optimizer: A Novel Physics-Inspired Metaheuristic Optimization Algorithm

Mohamed Abdel-Basset ^{1,*}, Reda Mohamed ¹, Karam M. Sallam ¹  and Ripon K. Chakraborty ² ¹ Faculty of Computers and Informatics, Zagazig University, Zagazig 44519, Sharqiyah, Egypt² School of Engineering and IT, UNSW Canberra at ADFA 2600, Campbell, ACT 2610, Australia

* Correspondence: mohamedbasset@zu.edu.eg

Abstract: This paper introduces a novel physical-inspired metaheuristic algorithm called “Light Spectrum Optimizer (LSO)” for continuous optimization problems. The inspiration for the proposed algorithm is the light dispersions with different angles while passing through rain droplets, causing the meteorological phenomenon of the colorful rainbow spectrum. In order to validate the proposed algorithm, three different experiments are conducted. First, LSO is tested on solving CEC 2005, and the obtained results are compared with a wide range of well-regarded metaheuristics. In the second experiment, LSO is used for solving four CEC competitions in single objective optimization benchmarks (CEC2014, CEC2017, CEC2020, and CEC2022), and its results are compared with eleven well-established and recently-published optimizers, named grey wolf optimizer (GWO), whale optimization algorithm (WOA), and salp swarm algorithm (SSA), evolutionary algorithms like differential evolution (DE), and recently-published optimizers including gradient-based optimizer (GBO), artificial gorilla troops optimizer (GTO), Runge–Kutta method (RUN) beyond the metaphor, African vultures optimization algorithm (AVOA), equilibrium optimizer (EO), grey wolf optimizer (GWO), Reptile Search Algorithm (RSA), and slime mold algorithm (SMA). In addition, several engineering design problems are solved, and the results are compared with many algorithms from the literature. The experimental results with the statistical analysis demonstrate the merits and highly superior performance of the proposed LSO algorithm.

Keywords: metaheuristic; optimization; physics-based algorithm; Light Spectrum Optimizer**MSC:** 68-04; 68Q25; 68T20; 68W25; 68W40; 68W50

Citation: Abdel-Basset, M.; Mohamed, R.; Sallam, K.M.; Chakraborty, R.K. Light Spectrum Optimizer: A Novel Physics-Inspired Metaheuristic Optimization Algorithm. *Mathematics* **2022**, *10*, 3466. <https://doi.org/10.3390/math10193466>

Academic Editor: Petr Stodola

Received: 2 August 2022

Accepted: 17 September 2022

Published: 23 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 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

The practical implications of metaheuristic algorithms have been widely spread, especially in the last few years. The reason behind this is the rapidity, high-quality solutions, and problem-independent characteristics of metaheuristics [1–5]. Unfortunately, no metaheuristics can efficiently solve all types of optimization problems. Consequently, a significant number of metaheuristics have been proposed from time to time, aiming to find efficient metaheuristics that are proper for various types of optimization problems. In particular, metaheuristics depend on the progress or movement behavior of a specified phenomenon or creature. By simulating such a progress or movement style, a metaheuristic can invade the search space of a problem as the environment of the simulated phenomenon or creature.

Metaheuristics depend on two search mechanisms while trying to find the best solution to the given problem. The first mechanism is exploration, which invades the unvisited search area. The second mechanism is exploitation, which searches around the found best solution [6]. The main factor of any metaheuristic success is balancing these two mechanisms. In particular, using more exploration makes metaheuristics unable to reach the global best solution. Alternatively, using more exploitation may lead to trapping into the local optima. In general, metaheuristics' searching mechanisms have stemmed from natural phenomena or the behavior of creatures.

Metaheuristics can be categorized based on their metaphors into seven main categories [7–11]: evolution-based, swarm-based, physics-based, human-based, chemistry-based, math-based, and others (See Figure 1). Evolution-based metaheuristics mimic the natural evolution process, which consists of select, crossover, and mutation processes such as genetic algorithm (GA) [12], genetic programming (GP) [13], evolution strategy (ES) [14], probability-based incremental learning (PBIL) [15], and differential evolution (DE) [16].

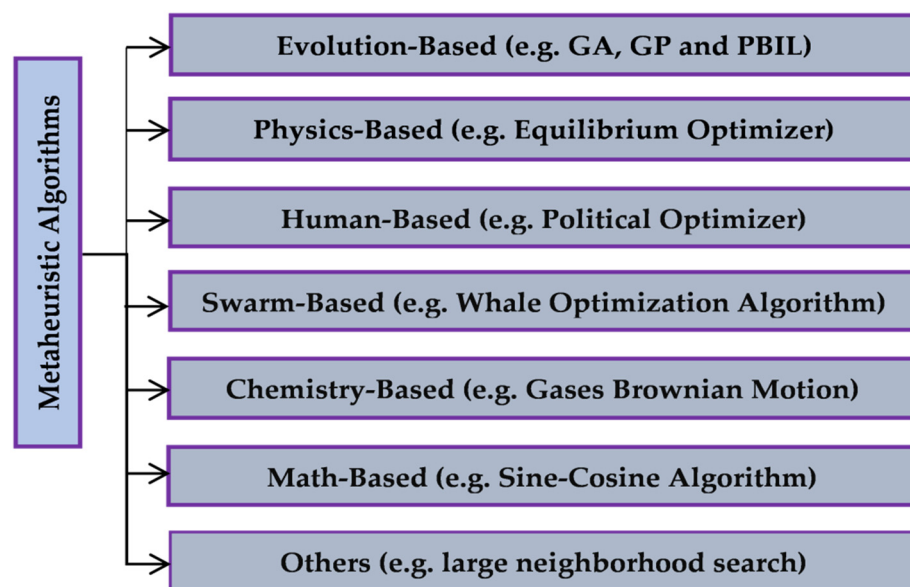


Figure 1. Categories of metaheuristic algorithms.

The second category, referred to as swarm-based algorithms, imitates the social behavior of swarms, birds, insects, and animal groups [17]. Some of the well-established and recently-published algorithms in this category are particle swarm optimization (PSO) [18], cuckoo search (CS) algorithm [19], flower pollination algorithm (FPA) [20], marine predators algorithm (MPA) [21], Harris hawks optimization (HHO) [22], salp swarm algorithm (SSA) [23], red fox optimizer (RFO) [24], duck swarm algorithm [25], chameleon swarm algorithm [26], artificial gorilla troops optimizer [27], cat optimization algorithm [28], donkey and smuggler optimization algorithm [29], krill herd algorithm [30], elephant herding optimization [31], wolf pack search algorithm [32], hunting search [33], monkey search [34], chicken swarm optimization [35], horse herd optimization algorithm (HOA) [36], moth search (MS) algorithm [37], earthworm optimization algorithm (EWA) [38], monarch butterfly optimization (MBO) [39], slime mold algorithm (SMA) [40] and whale optimization algorithm (WOA) [41]. In general, swarm-based metaheuristics have some advantages over evolution-based ones. In particular, a swarm-based metaheuristics search in a cumulative form preserves the information of subsequent search iterations. On the other hand, evolution-based metaheuristics ignore previous search information once the new population is generated. Additionally, evolution-based metaheuristics usually need more parameters than swarm-based metaheuristics. This makes swarm-based metaheuristics more applicable than evolution-based metaheuristics in most cases.

The third category of metaheuristics is human-based algorithms, which mimic human behaviors and human interactions in societies. The most popular algorithms belonging to this category are teaching–learning-based Optimization (TLBO) [42], harmony search (HS) [43], past present future (PPF) [44], political optimizer (PO) [45], brain storm optimization (BSO) [46], exchange market algorithm (EMA) [47], league championship algorithm (LCA) [48], poor and rich optimization algorithm [49], driving training-based optimization [50], gaining–sharing knowledge-based algorithm (GSK) [51], imperialist competitive algorithm (ICA) [52], and soccer league competition (SLC) [53].

The fourth category is physics-based algorithms, which are inspired by physical laws, such as inertia, electromagnetic force, gravitational force, and so on. In this category, the algorithms are based on physical principles to enable the search agents to interact and navigate the optimization problems' search space to reach the near-optimal solution. This category includes several algorithms like simulated annealing (SA) [54], gravitational search algorithm (GSA) [55], charged system search (CSS) [56], big-bang big-crunch (BBBC) [57], artificial physics algorithm (APA) [58], galaxy-based search algorithm (GbSA) [59], black hole (BH) algorithm [60], river formation dynamics (RFD) algorithm [61], henry gas solubility optimization (HGSO) algorithm [62], curved space optimization (CSO) [63], central force optimization (CFO) [64], water cycle algorithm (WCA) [65], water waves optimization (WWO) [66], ray optimization (RO) algorithm [67], gravitational local search algorithm (GLSA) [68], small-world optimization algorithm (SWOA) [69], multi-verse optimizer (MVO) [70], intelligent water drops (IWD) algorithm [71], integrated radiation algorithm (IRA) [72], space gravitational algorithm (SGA) [73], ion motion algorithm (IMA) [74], electromagnetism-like algorithm (EMA) [75], equilibrium optimizer (EO) [76], light ray optimization (LRO) [77], and Archimedes optimization algorithm (AOA) [78]. Both light ray optimization (LRO) [77] and ray optimization (RO) [67] simulate the reflection and refraction of the light rays, respectively, when transferred from a medium to a darker one, which is completely different from the proposed algorithm, referred to as Light Spectrum Optimizer (LSO), as illustrated later.

The chemistry-based metaheuristic algorithms in the fifth category are inspired by mimicking certain chemical laws; some of those algorithms are gases Brownian motion optimization (GBMO) [79], artificial chemical reaction optimization algorithm (ACROA) [80], and several others [81]. The sixth category, called math-based metaheuristics, is based on presenting metaheuristic algorithms inspired by simulating certain mathematics functions like the golden sine algorithm (GSA) [82], base optimization algorithm (BOA) [83], and sine-cosine algorithm [84]. Table 1 presents the category and inspiration of some of the recently-published metaheuristic algorithms—specifically published over the last three years.

The last category (Others) includes all the metaheuristic algorithms, which has not been inspired by the behaviors of creatures or natural phenomena, such as adaptive large neighborhood search technique (ALNS) [85], large neighborhood search (LNS) [86,87], and greedy randomized adaptive search procedure (GRASP) [88,89]. For example, the large neighborhood search technique is a metaheuristic algorithm based on improving an initial solution using destroy and repair operators.

Over the last few decades, several metaheuristic algorithms have been proposed, but unfortunately, most of these algorithms are not able to adapt themselves when tackling several optimization problems with various characteristics. Therefore, this paper proposes a novel physical-based metaheuristic algorithm called Light Spectrum Optimizer (LSO) for global optimization over a continuous search space. This novel metaheuristic is inspired by the sunlight ray dispersion while passing through the rain droplets causing the sparkle rainbow phenomenon. In particular, the mathematical formulation of the sunlight ray's reflection, refraction, and dispersion can be efficiently utilized for presenting a variety in the updating process to preserve the population diversity, in addition to accelerating the convergence speed when applied to different optimization problems. Experimentally, LSO is extensively assessed using several mathematical benchmarks like CEC2005, CEC2014, CEC2017, CEC2020, and CEC2022 to reveal its performance compared to several well-established metaheuristic algorithms. In addition, LSO is employed to solve some engineering design problems to further affirm its efficiency. The main advantages of the proposed metaheuristic are:

- Simple representation.
- Robustness.
- Balancing between exploration and exploitation.
- High-quality solutions.
- Swarm intelligence powerfulness.
- Low computational complexity.
- High scalability.

Table 1. Classification and inspiration of some recently-published metaheuristic algorithms.

Algorithm	Inspiration	Category	Year
Starling murmuration optimizer (SMO) [90]	Starlings' behaviors	Swarm-based	2022
Snake optimizer (SO) [91]	Mating behavior of snakes	Swarm-based	2022
Reptile Search Algorithm (RSA) [92]	Hunting behavior of Crocodiles	Swarm-based	2022
Archerfish hunting optimizer (AHO) [93]	Jumping behaviors of the archerfish	Swarm-based	2022
Water optimization algorithm (WAO) [94]	Chemical and physical properties of water molecules	Physics-based Chemistry-based	2022
Ebola optimization search algorithm (EOSA) [95]	Propagation mechanism of the Ebola virus disease	Others	2022
Beluga whale optimization (BWO) [96]	Behaviors of beluga whales	Swarm-based	2022
White Shark Optimizer (WSO)	Behaviors of great white sharks	Swarm-based	2022
Aphid–Ant Mutualism (AAM) [97]	The relationship between aphids and ants species is called Mutualism	Swarm-based	2022
Circle Search Algorithm (CSA) [98]	Geometrical features of circles	Math-based	2022
Pelican optimization algorithm (POA) [99]	The behavior of pelicans during hunting	Swarm-based	2022
Sheep flock optimization algorithm (SFOA) [100]	Shepherd and sheep behaviors in the pasture	Swarm-based	2022
Gannet optimization algorithm (GOA) [101]	Behaviors of gannets during foraging	Swarm-based	2022
Prairie dog optimization (PDO) [102]	The behavior of the prairie dogs	Swarm-based	2022
Driving Training-Based Optimization (DTBO) [50]	The human activity of driving training	Human-based	2022
Stock exchange trading optimization (SETO) [103]	The behavior of traders and stock price changes	Human-based	2022
Archimedes optimization algorithm (AOA) [78]	Archimedes law	Physics-based	2021
Golden eagle optimizer (GEO) [104]	Golden eagles' hunting process	Swarm-based	2021
Heap-based optimizer (HBO) [105]	Corporate rank hierarchy	Human-based	2021
African vultures optimization algorithm (AVOA) [106]	African vultures' lifestyle	Swarm-based	2021
Artificial gorilla troops optimizer (GTO) [27]	Gorilla troops' social intelligence	Swarm-based	2021
Quantum-based avian navigation optimizer algorithm (QANA) [107]	Migratory birds' navigation behaviors	Evolution-based (Based DE)	2021
Colony predation algorithm (CPA) [108]	Corporate predation of animals	Swarm-based	2021
Lévy flight distribution (LFD) [42]	Lévy flight random walk	Physics-based	2020
Political Optimizer (PO) [45]	Multi-phased process of politics	Human-based	2020
Marine predators algorithm (MPA) [21]	Foraging strategy in the ocean between predators and prey	Swarm-based	2020
Equilibrium optimizer (EO) [76]	Mass balance models	Physics-based	2020

These advantages are proved with three different validating experiments that include several optimization problems with various characteristics. Besides, LSO is compared with

many other optimization algorithms, and the results are analyzed with the appropriate statistical tests. The experimental findings affirm the superiority of LSO compared to all the other rival algorithms. Finally, the main contributions of this study are listed as follows:

- Proposing a novel physical-based metaheuristic algorithm called Light Spectrum Optimizer (LSO), inspired by the sparkle rainbow phenomenon caused by passing sunlight rays through the rain droplets.
- Validating LSO using four challengeable mathematical benchmarks like CEC2014, CEC2017, CEC2020, and CEC2022, as well as several engineering design problems.
- The experimental findings, along with the Wilcoxon rank-sum test as a statistical test, illustrate the merits and highly superior performance of the proposed LSO algorithm

The remainder of this work is organized as follows. Section 2 gives the background illustration of the inspiration and the mathematical modelling of the rainbow phenomenon. Section 3 explains the mathematical formulation and the searching procedure of LSO. In Section 4, various experiments are done on the CEC2005, CEC2014, CEC2017, CEC2020, and CEC2022 benchmarks, and their experimental results are analyzed with the proper statistical analysis. Additionally, LSO sensitivity is presented. In Section 5, popular engineering design problems are solved with LSO.

2. Background

Rainbow is one of the most fabulous metrological wonders. From the physical perspective, it is a half-circle of spectrum colors created by dispersion and internal reflection of sunlight rays that hit spherical rain droplets [109]. When a white ray hits a water droplet, it changes its direction by refracting and reflecting inside and outside the droplet (sometimes more than once) [110]. In other words, the rainbow is formed by light rays' refraction, reflection, and dispersion through water droplets.

According to Descartes's laws [111,112], refraction occurs when the light rays travel from one material to another with a different refractive index. When light rays hit the outer surface of a droplet, some light rays reflect away from the droplet while the others are refracted. The refracted light rays hit the inner surface of a droplet, causing another reflection and refracting away from the droplet with different angles, which causes the white sunlight to be dispersed into its seven spectral colors: red, orange, yellow, green, blue, indigo, and violet, as depicted in Figure 2. These spectral colors, differs according to the angles of deviations, which range from 40° (violet) to 42° (red) [113,114] (See Figure 2).

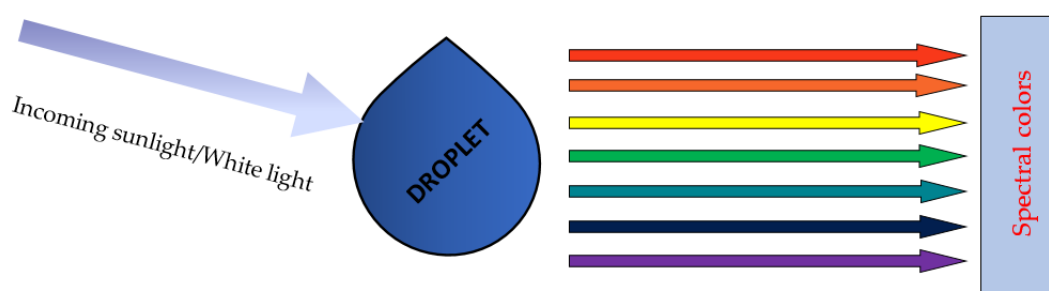


Figure 2. Light dispersion and colors of rainbow.

Mathematically, the refraction and reflection of the rainbow spectrum have been illustrated by Snell's laws. Snell's law said that the ratio between the sines of the incident and refracted angles is equal to the ratio between the refractive indices of air and water, as [115]:

$$\frac{\sin(\theta_1)}{\sin(\theta_2)} = \frac{k_2}{k_1} \quad (1)$$

where θ_1 is the incident angle, θ_2 is the refracted angle, k_2 is the refractive index of water, and k_1 is the refractive index of air.

In this work, Snell's law is used in its vector form. As shown in Figure 3, all normal, incidents, refracted, and reflected rays are converted to vectors. The mathematical formulation of the refracted ray can be expressed as [116]:

$$L_1 = \frac{1}{k} [L_0 - n_A(n_A \cdot L_0)] - n_A \left[1 - \frac{1}{k^2} + \frac{1}{k^2} (n_A \cdot L_0)^2 \right]^{\frac{1}{2}} \quad (2)$$

where L_1 is the refracted light ray, k is the refractive index of the droplet, L_0 is the incident light ray, and n_A is the normal line at the point of incidence. Meanwhile, the inner reflected ray can be formulated as:

$$L_2 = L_1 - 2n_B(n_B \cdot L_1) \quad (3)$$

where L_2 is the inner reflected light ray and n_B is the normal line at the point of inner reflection. Finally, the outer refracted ray is expressed as:

$$L_3 = k[L_2 - n_C(n_C \cdot L_2)] + n_C[1 - k^2 + k^2(n_C \cdot L_2)^2]^{\frac{1}{2}} \quad (4)$$

where L_3 is the outer refracted light ray and n_C is the normal line at the point of outer refraction.

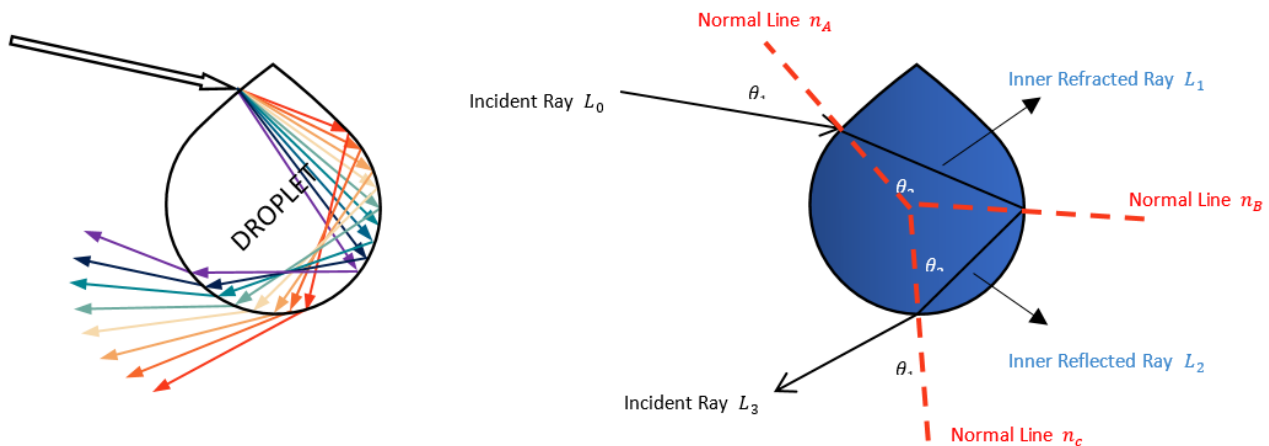


Figure 3. Light dispersion and colors of rainbow and vector form of refraction and reflection in rainbow.

3. Light Spectrum Optimizer (LSO)

As discussed before, the rainbow spectrum rays are caused by colorful light dispersion. In this paper, the proposed algorithm takes its inspiration from this metrological phenomenon. In particular, LSO is based on the following assumptions:

- (1) Each colorful ray represents a candidate solution.
- (2) The dispersion of light rays ranges from 40° to 42° or have a refractive index that varies between $k^{red} = 1.331$ and $k^{violet} = 1.344$.
- (3) The population of light rays has a global best solution, which is the best dispersion reached so far.
- (4) The refraction and reflection (inner or outer) are randomly controlled.
- (5) The current solution's fitness value controls a colorful rainbow curve's first and second scattering phases compared to the best so-far solution's fitness. Suppose the fitness value between them is so close. In that case, the algorithm will apply the first scattering phase to exploit the regions around the current solution because it might be so close to the near-optimal solution. Otherwise, the second phase will be applied to help the proposed algorithm avoid getting stuck in the regions of the best-so-far solution because it might be local minima.

Next, the detailed mathematical formulation of LSO will be discussed.

3.1. Initialization Step

The search process of LSO begins with the random initialization of the initial population of white lights as:

$$\vec{x}^0 = lb + RV_1(ub - lb) \quad (5)$$

where $\vec{x}^0$ is the initial solution, RV_1 is a vector of uniform random numbers generated between $[0, 1]$ with a length equal to the given problem dimension (d), and lb and ub are the lower and upper bounds of the search space, respectively. After that, the generated initial solutions are evaluated in order to determine the global and personal best solutions.

3.2. Colorful Dispersion of Light Rays

In this subsection, we discuss the mathematical formulation of rainbow spectrum directions, colorful rays scattering, and the exploration and exploitation mechanisms of LSO.

3.2.1. The Direction of Rainbow Spectrums

After the initialization, the normal vector of inner refraction $\vec{x}_{nA}$, inner reflection $\vec{x}_{nB}$, and outer refraction $\vec{x}_{nC}$ are calculated as:

$$\vec{x}_{nA} = \frac{\vec{x}_t^r}{\text{norm}(\vec{x}_t^r)} \quad (6)$$

$$\vec{x}_{nB} = \frac{\vec{x}_t^p}{\text{norm}(\vec{x}_t^p)} \quad (7)$$

$$\vec{x}_{nC} = \frac{\vec{x}^*}{\text{norm}(\vec{x}^*)} \quad (8)$$

where $\vec{x}_t^r$ is a randomly selected solution from the current population at iteration t , $\vec{x}_t^p$ is the current solution at iteration t , $\vec{x}^*$ is the global best solution ever founded, and $\text{norm}(\cdot)$ indicates the normalized value of a vector and computed according to the following formula:

$$\text{norm}(\vec{x}) = \sqrt{\sum_{j=0}^d x_j^2} \quad (9)$$

where d stands for the number of dimensions in an optimization problem. $\vec{x}$ is the input vector to the norm function to normalize it. x_j is the j th dimension in the input vector $\vec{x}$. For the incident light ray, it is calculated as follows:

$$X_{\text{mean}} = \frac{\sum_i^N \vec{x}_i}{N} \quad (10)$$

$$\vec{x}_{L0} = \frac{X_{\text{mean}}}{\text{norm}(X_{\text{mean}})} \quad (11)$$

where $\vec{x}_{L0}$ is the incident light ray, X_{mean} is the mean of the current population of solutions $\vec{x}_i (i = 1, \dots, N)$, and N is the population size.

Then, the vectors of inner and outer refracted and reflected light rays are calculated as:

$$\vec{x}_{L1} = \frac{1}{k^r} [\vec{x}_{L0} - \vec{x}_{nA} (\vec{x}_{nA} \cdot \vec{x}_{L0})] - \vec{x}_{nA} \left| 1 - \frac{1}{(k^r)^2} + \frac{1}{(k^r)^2} (\vec{x}_{nA} \cdot \vec{x}_{L0})^2 \right|^{\frac{1}{2}} \quad (12)$$

$$\vec{x}_{L2} = \vec{x}_{L1} - 2\vec{x}_{nB} (\vec{x}_{L1} \cdot \vec{x}_{nB}) \quad (13)$$

$$\vec{x}_{L3} = k^r [\vec{x}_{L2} - \vec{x}_{nC} (\vec{x}_{nC} \cdot \vec{x}_{L2})] + \vec{x}_{nC} \left| 1 - (k^r)^2 + (k^r)^2 (\vec{x}_{nC} \cdot \vec{x}_{L2})^2 \right|^{\frac{1}{2}} \quad (14)$$

where $\vec{x}_{L1}$, $\vec{x}_{L2}$, and $\vec{x}_{L3}$ are the inner refracted, inner reflected, and outer refracted light rays, respectively. k^r stands for the refractive index, which is updated randomly between k^{red} and k^{violet} to define a random spectrum color as:

$$k^r = k^{red} + RV_1 (k^{violet} - k^{red}) \quad (15)$$

where RV_1 is a uniform random number generated randomly between $[0, 1]$.

Table 2 presents a numerical example to illustrate the vectors with six dimensions generated by the previously described equations with noting that the vectors $\vec{x}_t^r$, $\vec{x}_t^p$, and $\vec{x}^*$ presented at the same table are randomly generated between 100 and -100 . The values of the inner refracted and inner reflected are obvious from this table. Outer refracted vectors could not be employed alone to update the individuals, which have to be ranged between 100 and -100 because the change rate in the updated solutions will be so low. Hence, many function evaluations will be consumed to reach better solutions. Therefore, the equations described in the next section are adapted to deal with this problem by extensively encouraging the exploration operator of the newly-proposed algorithm.

Table 2. An illustrative numerical example to the results generated according to Equations (6)–(14).

	x_1	x_2	x_3	x_4	x_5	x_6
$\vec{x}_t^r$	61.4064	−77.7085	−91.5639	−70.6667	−41.4573	−58.8625
$\vec{x}_t^p$	−33.8424	85.3758	−13.0301	−37.5489	62.1643	39.1392
$\vec{x}^*$	−2.2651	25.3057	76.0485	45.3216	−75.5683	−80.4718
$\vec{x}_{nA}$	0.1006	−0.1273	−0.1500	−0.1157	−0.0679	−0.0964
$\vec{x}_{nB}$	−0.0622	0.1569	−0.0239	−0.0690	0.1143	0.0719
$\vec{x}_{nC}$	−0.0047	0.0529	0.1590	0.0947	−0.1580	−0.1682
$\vec{x}_{L0}$	−0.0090	0.0972	0.0282	−0.0217	−0.0122	−0.0749
$\vec{x}_{L1}$	−0.0989	0.1893	0.1586	0.0900	0.0532	0.0325
$\vec{x}_{L2}$	−0.0765	0.1327	0.1672	0.1149	0.0119	0.0066
$\vec{x}_{L3}$	−0.1054	0.2106	0.3229	0.2128	−0.0823	−0.0959

3.2.2. Generating New Colorful Ray: Exploration Mechanism

After the calculation of the rays' directions, we calculate the candidate solutions according to the value of a randomly generated probability between 0 and 1, referred to as p . In particular, if the value of p is lower than a number generated randomly between 0 and 1, then the new candidate solution will be calculated as:

$$\vec{x}_{t+1} = \vec{x}_t + \epsilon RV_1^n GI(\vec{x}_{L1} - \vec{x}_{L3}) \times (\vec{x}_{r1} - \vec{x}_{r2}) \quad (16)$$

Otherwise, the new candidate solution will be calculated as:

$$\vec{x}_{t+1} = \vec{x}_t + \epsilon RV_2^n GI(\vec{x}_{L2} - \vec{x}_{L3}) \times (\vec{x}_{r3} - \vec{x}_{r4}) \quad (17)$$

where $\vec{x}_{t+1}$ is the newly generated candidate solution, $\vec{x}_t$ is the current candidate solution at iteration t . $r1, r2, r3$, and $r4$ are indices of four solutions selected randomly from the current population. RV_1^n and RV_2^n are vectors of uniform random numbers that are generated between $[0, 1]$. ϵ is a scaling factor that is calculated using (18). GI is an adaptive control factor based on the inverse incomplete gamma function and computed according to (19).

$$\epsilon = a \times RV_3^n \quad (18)$$

where RV_3^n a vector of normally distributed random numbers with a mean equal to zero and a standard deviation equal to one, and a is an adaptive parameter that can be calculated using (20).

$$GI = a \times r^{-1} \times P^{-1}(a, 1) \quad (19)$$

GI is an adaptive control factor. r is a uniform random number between $[0, 1]$ that is inversed to promote the exploration operator throughout the optimization process because inverting any random value will generate a new decimal number greater than 1, which might take the current solution to far away regions within the search space for finding a better solution. P^{-1} is the inverse incomplete gamma function for the corresponding value of a .

$$A = RV_2 \left(1 - \left(\frac{t}{T_{max}} \right) \right) \quad (20)$$

where t is the current iteration number, RV_2 is a scalar numerical value of uniform random numbers generated between $[0, 1]$, and T_{max} is the maximum number of function evaluations.

When the input numbers are greater than 0.5, this inverse incomplete gamma function generates high numerical values starting from almost 0.8 and ending at nearly 5.5, as described in Figure 4; otherwise, it generates decimal values down to 0. When the input numbers to this function are high, it will encourage the exploration operator. However, the highly-generated value might take the updated solutions out of the search boundary, and hence the algorithm might be converted into a randomization process because the boundary checking method will move those infeasible solutions back again into the search space. Therefore, the factor, a , described before in (20), is modeled with the inverse incomplete gamma values to reduce their magnitude and avoid the randomization process when the input values are high. Both the inverse function and factor a decrease gradually with increasing the current iteration, and hence the optimization process will be gradually converted from the exploration operator into the exploitation that might lead to falling into local minima. Therefore, to support the exploration operator throughout the optimization process, the inverse of a number generated randomly between 0 and 1 will be modeled with both the inverse function and factor a as defined in (19).

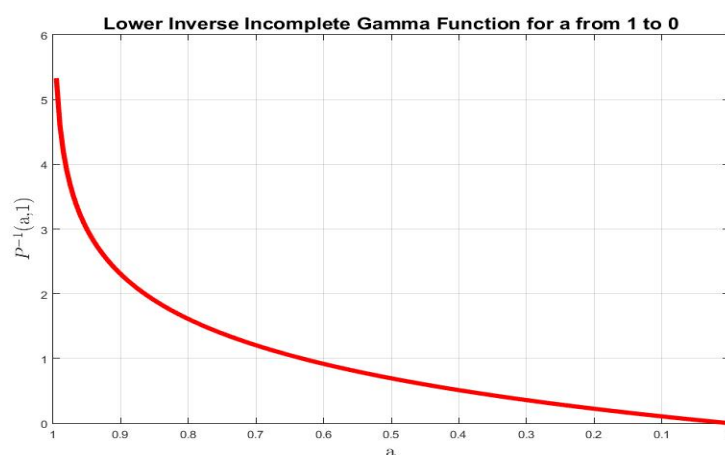


Figure 4. The behavior of the inverse incomplete gamma function with respect to a values.

3.2.3. Colorful Rays Scattering: Exploitation Mechanism

This phase helps to scatter the rays in the direction of the best-so-far solution, the current solution, and a solution selected randomly from the current population to improve its exploitation operator. At the start, the algorithm will scatter the rays around the current solution to exploit the region around it to reach better outcomes. However, this might reduce the convergence speed of LSO, so an additional step size applied with a predefined probability β is integrated to move the current solution in the direction of the best-so-far solution to overcome this problem. The mathematical model of scattering around the current solution is as follows:

$$\vec{x}_{t+1} = \vec{x}_t + RV_3 \times (\vec{x}_{r1} - \vec{x}_{r2}) + RV_4^n \times (R < \beta) \times (\vec{x}^* - \vec{x}_t) \quad (21)$$

where $\vec{x}^*$ is the best-so-far solution, and $\vec{x}_{r1}$ and $\vec{x}_{r2}$ are two solutions selected randomly from the current population. RV_3 includes a number selected randomly at the interval of 0 and 1. RV_4^n is a vector including numbers generated randomly between 0 and 1. The second scattering phase is based on generating rays in a new position based on the best-so-far solution and the current solution according to the following formula:

$$\vec{x}_{t+1} = 2 \cos(\pi \times r_1)(\vec{x}^*)(\vec{x}_t) \quad (22)$$

where r_1 is a randomly generated numerical value at the interval of 0 and 1. π indicates the ratio of the perimeter of a circle to its diameter. Exchanging between the first and second scattering phases is achieved based on a predefined probability P_e as shown in the following formula:

$$\vec{x}_{t+1} = \begin{cases} \text{Eq. (21)} & \text{if } R < P_e \\ \text{Eq. (22)} & \text{Otherwise} \end{cases} \quad (23)$$

where R is a number generated randomly between 0 and 1. The last scattering phase is based on generating a new solution according to a solution selected randomly from the population and the current solution according to the following formula:

$$\vec{x}_{t+1} = \left(\vec{x}_{r1}^p + |RV_5| \times (\vec{x}_{r2} - \vec{x}_{r3}) \right) \times \vec{U} + (1 - \vec{U}) \times \vec{x}_t \quad (24)$$

where RV_5 is a scalar value of normally distributed random numbers with a mean equal to zero and a standard deviation equal to one, and $\vec{U}$ is a vector including random values of 0 and 1. $|\dots|$ is the absolute symbol, which converts the negative values into positive ones and returns the positive numbers as passed. Exchanging between Equations (23) and (24) is based on computing the difference between the fitness value of each solution and that of the best-so-far solution and normalizing this difference between 0 and 1 according to (25). If this difference is less than a threshold value R_1 generated randomly between 0 and 1, (23) will be applied; otherwise, (24) is applied. Our hypothesis is based herein computing the probabilistic fitness value using (25) to determine how far the current light ray is close to the best-so-far light ray. If the probabilistic fitness value for the i th light ray is smaller than R_1 , it is preferable to scatter this light ray in the same direction as the best-so-far solution. Our proposed algorithm suggests this hypothesis to maximize its performance when dealing with any optimization problem that needs a high-exploitation operator to accelerate the convergence speed and save computational costs.

$$F' = \left| \frac{F - F_b}{F_b - F_w} \right| \quad (25)$$

where F , F_b , and F_w indicate the fitness values of the current solution, best-so-far solution, and worst solution, respectively. However, the probability of applying (23) when the value of F' is high is a little. For example, Figure 5 has tracked the values of F' for an agent and

the random number R_1 during the optimization process of a test function; this figure shows that the F' values are nearly greater than R_1 for most of the optimization process, as shown by red points that are much greater than the blue points in the 9 subgraphs depicted in Figure 5, and thus the chance of firing the first and second scattering stages is so low when relying solely on factor F' . Therefore, exchanging between (23) and (24) is also applied with a predefined probability P_s to further promote the first and second scattering stages for accelerating convergence toward the best-so-far solution. Finally, exchanging between these two equations is formulated in the following formula:

$$x_{t+1}^{\rightarrow} = \begin{cases} \text{Eq. (23)} & \text{if } R < P_s \mid F' < R_1 \\ \text{Eq. (24)} & \text{Otherwise} \end{cases} \quad (26)$$

where R and R_1 are numbers generated randomly between 0 and 1.

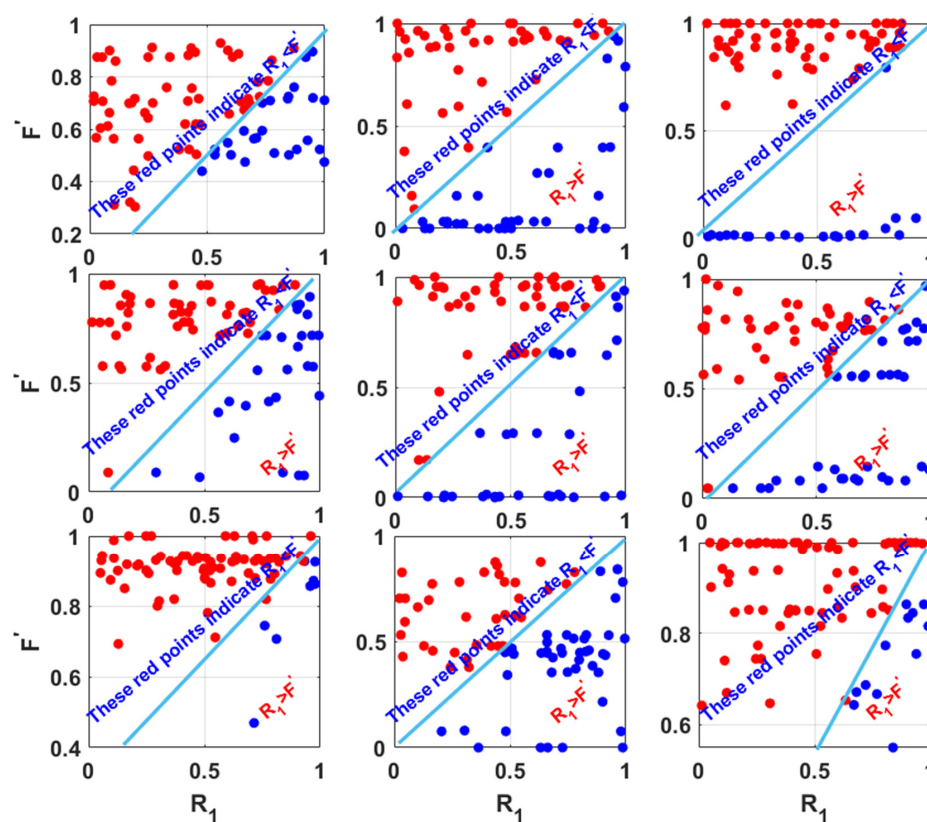


Figure 5. Tracing F' 's values versus R_1 for an individual over 9 independent runs: The red points indicate that $R_1 < F'$; and the blue points indicate that $R_1 > F'$.

3.3. LSO Pseudocode

The pseudocode of the proposed algorithm is stated in Algorithm 1, and the same steps are depicted in Figure 6. Some solutions might go outside the problem's search space. Thus, they have to be returned back into the search space to find feasible solutions to the problem. There are two common ways to convert the infeasible solutions, which go outside the search space to feasible ones; the first one is based on setting the lower bound to the dimensions, which are smaller, and setting the upper bound to these, which are higher; while the second one is based on generating new random values within the search boundaries of the dimensions, which go outside the search space of the problem. Within our proposed algorithm, we make hybridization between these two methods to improve the convergence rate by the first and the exploration by the second. This hybridization is achieved based on a predefined probability P_h , which is estimated within the experiments section.

Algorithm 1: LSO Pseudo-Code

Input: Population size of light rays N , problem Number of Iterations $Tmax$
Output: The best light dispersion x^* and its fitness
 Generate initial random population of light rays x_i ($i = 1, 2, 3, \dots, N$)
 $t = 0$

```

1  While ( $t < Tmax$ )
2      for each light ray
3          evaluate the fitness value
4           $t = t + 1$ 
5          keep the current global best  $x^*$ 
6          Update the current solution if the updated solution is better.
7          determine normal lines  $\vec{x}_{nA}$ ,  $\vec{x}_{nB}$ , &  $\vec{x}_{nC}$ 
8          determine direction vectors  $\vec{x}_{L0}$ ,  $\vec{x}_{L1}$ ,  $\vec{x}_{L2}$ , &  $\vec{x}_{L3}$ 
9          update the refractive index  $k'$ 
10         update  $a$ ,  $\epsilon$ , and  $GI$ 
11         Generate two random numbers:  $p, q$  between 0 and 1
12         %%%Generating new ColorFul ray: Exploration phase
13         if  $p \leq q$ 
14             update the next light dispersion using Equation (16)
15         Else
16             update the next light dispersion using Equation (17)
17         end if
18         evaluate the fitness value
19          $t = t + 1$ 
20         keep the current global best  $x^*$ 
21         Update the current solution if the updated solution is better.
22         %%%Scattering phase: exploitation phase
23         Update the next light dispersion using Equation (26)
24     end for
25 end while
26 Return  $x^*$ 
  
```

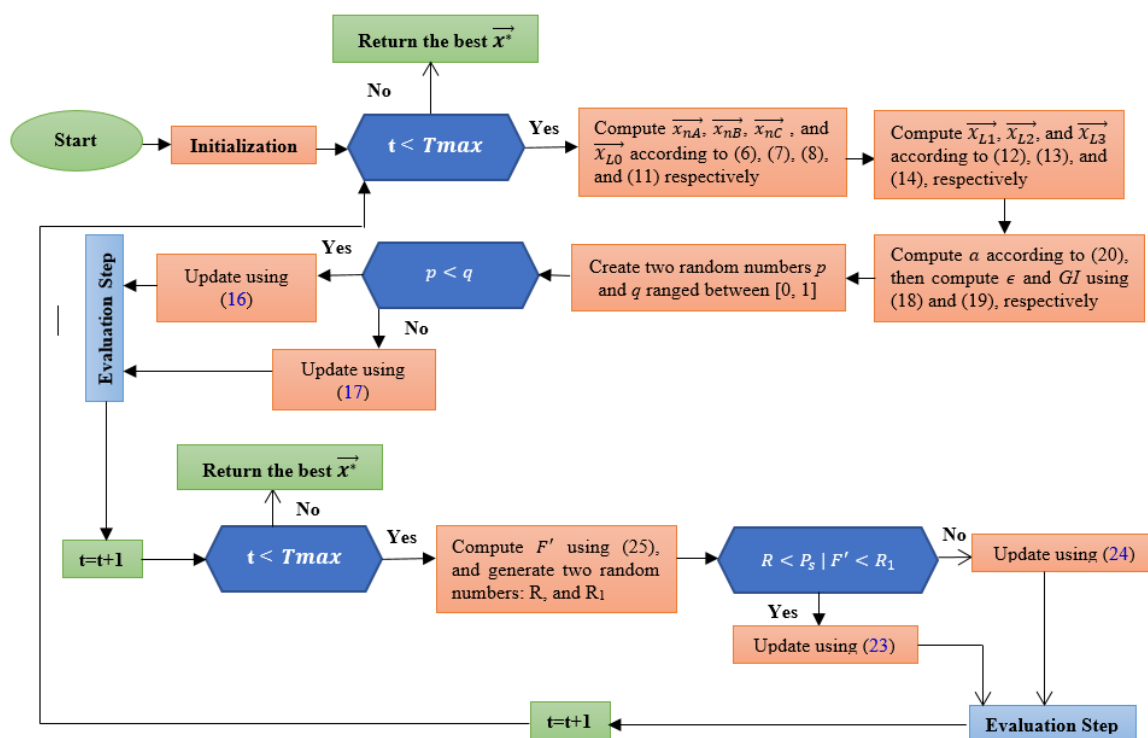


Figure 6. Flowchart of LSO.

3.4. Searching Behavior and Complexity of LSO

In this section, we will discuss the searching schema of LSO and its computational complexity.

A. Searching behavior of LSO

As discussed before, LSO reciprocates the methods of finding the next solution through the use of $\vec{x}^*$, $\vec{x}_t^r$, and $\vec{x}_t^p$. In other words, $\vec{x}_{nA}$ is calculated according to a randomly selected solution, which ensures the exploration of the search space. Meanwhile, the calculation of $\vec{x}_{nB}$ and $\vec{x}_{nC}$ depends on the global and personal best solutions, respectively. This preserves the exploitation of the search space. Another exploitation consolidation is preserved by the usage of the inverse incomplete gamma function [117], which can be expressed as follows:

$$j = P^{-1}(y, w) \quad (27)$$

$$y = P(j, w) = \frac{1}{\Gamma(w)} \int_0^z t^{w-1} e^{-t} dt \quad (28)$$

where w is a scaling factor that is greater than or equal 0.

Figure 7 depicts the LSO's exploration and exploitation operators to illustrate the behavior of LSO experimentally. This figure is plotted by displaying the search core of an individual during the exploration and exploitation phases for the first two dimensions (X_1 and X_2). From this figure, specifically Figure 7a, which depicts the LSO's exploitation operator, it is obvious that this operator focuses its search toward a specific region, often the best-so-far region, to explore the solutions around and inside this region in the hope of reaching better solutions in a lower number of function evaluations. On the other side, Figure 7b pictures the exploration behavior of LSO to show how far LSO could reach; this figure shows that the individuals within the optimization process try to attack different regions, far from the current, within the search space of the optimization process for reaching the most promising region, which is attacked using the exploitation operator discussed formerly.

B. Space and Time Complexity

- (1) **LSO Space Complexity** The space complexity of any metaheuristic can be defined as the maximum space required during the search process. The big O notation of LSO space complexity can be stated as $O(N \times d)$, where N is the number of search agents, and d is the dimension of the given optimization problem.
- (2) **LSO Time Complexity** The time complexity of LSO is analyzed in this study using asymptotic analysis, which could analyze the performance of an algorithm based on the input size. Other than the input, all the other operations, like the exploration and exploitation operators, are considered constant. There are three asymptotic notations: big-O, omega, and theta, which are commonly used to analyze the running time complexity of an algorithm. The big-O notation is considered in this study to analyze the time complexity of LSO because it expresses the upper bound of the running time required by LSO for reaching the outcomes.

The time complexity of any metaheuristic depends on the required time for each step of the algorithm, like generating the initial population, updating candidate solutions, etc. Thus, the total time complexity is the sum of all such time measures. The time complexity of LSO results from three main algorithm steps:

- (1) Generation of the initial population.
- (2) Calculation of candidate solutions.
- (3) Evaluation of candidate solutions.

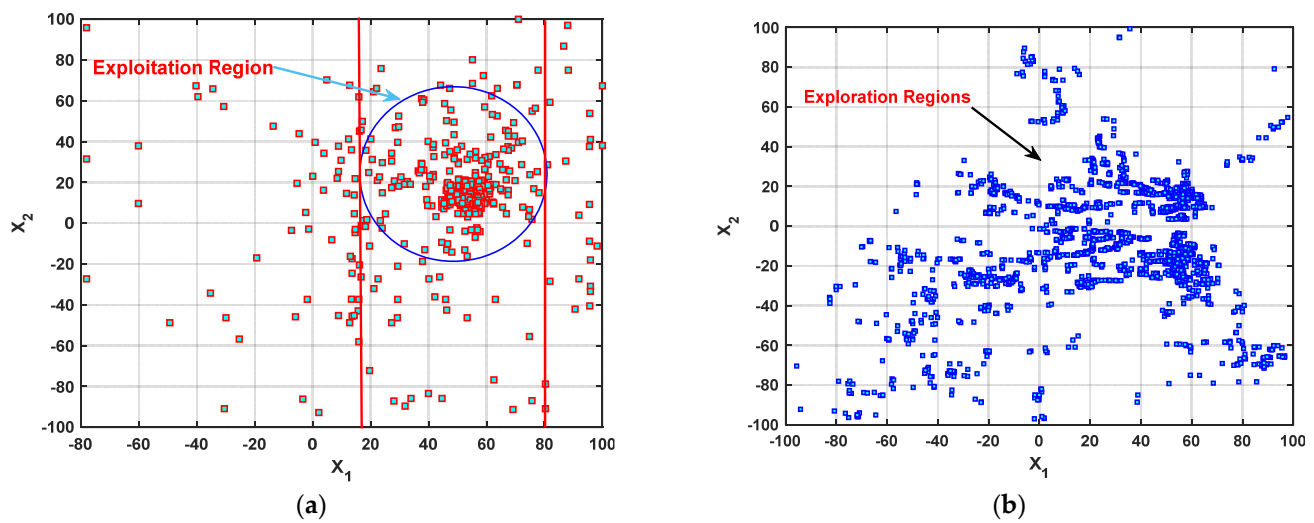


Figure 7. Depiction of exploration and exploitation stages of the proposed algorithm (LSO). (a) Exploitation phase. (b) Exploration phase.

The first initialization step has a time complexity equal to $O(N \times d)$. The candidate solutions calculation has a time complexity $O(T_{max} \times N \times d)$, which includes the evaluation of the generated solutions and updating the current best solution, Where T_{max} is the maximum number of search iterations. So, the total time complexity of LSO in big-O is $O(N \times d \times T_{max})$, which is confirmed in detail in Table 3.

Table 3. Execution time of each line in the proposed algorithm according to using the asymptotic analysis.

Algorithm 1: LSO Pseudo-Code		Execution Time for Each Line
Input: Population size of light rays N , problem Number of Iterations T_{max}		One execution time for each input. This line contains 3 inputs, and the total execution time will be equal to 3
Output: The best light dispersion x^* and its fitness Generate an initial random population of light rays x_i ($i = 1, 2, 3, \dots, N$)		- The output will be returned one time, so the execution time of this line will be 1 - Regarding the initialization, this stage will initialize d dimensions for N solutions, so the time complexity will be $O(N \times d)$
1	While ($t < T_{max}$)	This line will be executed T_{max} times
2	for each light ray x_i ($i = 1, 2, 3, \dots, N$)	Executed N times multiplied by T_{max} , which $O(N \times T_{max})$
3	evaluate the fitness value	$O(N \times d \times T_{max})$ since the objective function will observe each dimension in the updated solution
4	$t = t + 1$	$O(N \times T_{max})$
5	keep the current global best x^*	$O(N \times d \times T_{max})$
6	Update the current solution if the updated solution is better.	$O(N \times d \times T_{max})$
7	determine normal lines $\vec{x}_{nA}$, $\vec{x}_{nB}$, & $\vec{x}_{nC}$	$3 \times N \times d \times T_{max} = O(N \times d \times T_{max})$ 3 here indicates the number of generated vectors

Table 3. Cont.

Algorithm 1: LSO Pseudo-Code		Execution Time for Each Line
8	determine direction vectors $\vec{x}_{L0}$, $\vec{x}_{L1}$, $\vec{x}_{L2}$, & $\vec{x}_{L3}$	$3 \times N \times d \times Tmax = O(N \times d \times Tmax)$ 3 here indicates the number of generated direction vectors
9	update the refractive index k'	$O(N \times Tmax)$
10	update a , ϵ , & GI	$O(N \times Tmax)$
11	Generate two random numbers: p , between 0 and 1	$O(N \times Tmax)$
%%%Generating new ColorFul ray: Exploration phase		Comments not implemented
12	if $p \leq q$	$O(N \times Tmax)$
13	update the next light dispersion using Equation (16)	$O(N \times d \times Tmax)$
14	Else	$O(N \times Tmax)$
15	update the next light dispersion using Equation (17)	$O(N \times d \times Tmax)$
16	end if	$O(N \times Tmax)$
17	evaluate the fitness value	$O(N \times d \times Tmax)$
18	$t = t + 1$	$O(N \times Tmax)$
19	keep the current global best x^*	$O(N \times d \times Tmax)$
20	Update the current solution if the updated solution is better.	$O(N \times d \times Tmax)$
%%%Scattering phase: exploitation phase		Comments not implemented
21	Update the next light dispersion using Equation (25)	$O(N \times d \times Tmax)$
22	end for	$O(N \times Tmax)$
23	end while	$O(N \times Tmax)$
24	Return x^*	1 time
By Summing the execution time of all lines, it is obvious that the highest growth rate is $O(N \times d \times Tmax)$. Hence, the time complexity of LSO is $O(N \times d \times Tmax)$ since it has the highest growth rate		

3.5. Difference between LSO, RO, and LRO

This section compares the proposed algorithm to two other metaheuristic algorithms inspired by light reflection and refraction to demonstrate that LSO is completely different from those algorithms in terms of inspiration, formulation of candidate solutions, and the variation of the updating process, as illustrated in Table 4.

Table 4. Comparison between LSO, RO, and LRO.

Characteristics	LSO	RO	LRO
Inspiration	Simulating the light movement and orientation in the rainbow metrological phenomenon.	Simulating Snell's light refraction law when light transfers from a lighter medium to a darker medium.	Simulating the light's reflection and refraction.
Formulation	LSO mainly depends on the vector representation of the rainbow and its intersperse in the sky.	The formulation of RO depends on the general Snell's law of light ray transformation from a medium to a darker one to ray tracing in 2-dimensional and 3-dimensional spaces.	The updating of candidate solutions depends on the division of a search space into grid cells and then considering these cells as reflection and refraction points.

Table 4. Cont.

Characteristics	LSO	RO	LRO
Variation	- Variations in the updating process enable it to overcome several optimization problems of varying difficulty. - Having a strong exploration operator due to employing both the inverse incomplete gamma function and the inverse random number in an effective way. - Enjoying in strong exploitation operator aiding in exploiting the regions around the current solution, the best-so-far solution, and solutions selected randomly from the population.	- Having a weak exploration operator due to its inability to find the optimal solution for most fixed-dimensional multimodal optimization problems with many local minima necessitates using a strong exploration operator. On the contrary, our proposed algorithm can solve all fixed-dimensional multimodal problems. - Its performance for some unimodal problems requiring a strong exploitation operator is subpar.	- Its updating process is limited due to its reliance solely on refraction and reflection. - Slower convergence speed than LSO due to the weakness of its exploitation operator.

4. Experimental Results and Discussions

In this section, we investigate the efficiency of LSO by different benchmarks, including CEC2005, CEC2014, CEC2017, CEC2020, and CEC2022. In addition, the sensitivity and scalability analyses of the proposed algorithm are introduced in this section.

4.1. Benchmarks and Compared Optimizers

We first validate the efficiency of LSO by solving 20 classical benchmarks CEC2005 that were selected from [118–120]. The selected benchmarks consist of three classes: unimodal, multi-modal, and fixed-dimension multi-modal. Both unimodal and multimodal functions of CEC2005 are solved in 100 dimensions. Appendix A Tables A1–A7 shows the characteristics of these three classes, which are mathematical formulations of benchmarks, dimension (D), boundaries of the search space (B), and the global optimal solution (OS). Furthermore, the proposed algorithm is tested on solving challengeable benchmarks like CEC2014, CEC2017, CEC2020, and CEC2020, which are described in Appendix A Tables A1–A7. The dimensions of these challengeable benchmarks are set to 10. In addition, the Wilcoxon test [121] is performed to analyze both algorithms' performance during the 30 runs with a 5% significance level.

The experimental results of LSO are compared with highly-cited state-of-the-art optimization algorithms like grey wolf optimizer (GWO) [122], whale optimization algorithm (WOA) [123], and salp swarm algorithm (SSA) [23], evolutionary algorithms like differential evolution (DE), and recently-published optimizers including gradient-based optimizer (GBO) [124], artificial gorilla troops optimizer (GTO) [27], Runge–Kutta method (RUN) beyond the metaphor [125], African vultures optimization algorithm (AVOA) [106], equilibrium optimizer (EO) [76], grey wolf optimizer (GWO) [122], reptile search algorithm (RSA) [92], and slime mold algorithm (SMA) [40]. Both comparisons are based on standard deviation (SD), an average of fitness values (Avr), and rank. All the algorithms are coded in MATLAB® 2019. All experiments are performed on a 64-bit operating system with a 2.60 GHz CPU and 32 GB RAM. For a fair comparison, each algorithm runs for 25 independent times, the maximum number of function evaluations and population size are of 50,000 and 20, respectively (These parameters are constant within our experiments for all validated benchmarks). The other algorithms' parameters are kept as standard. The used parameters are given in Table 5.

Table 5. Parameters of the compared algorithms.

Algorithms	Parameters	Value	Algorithms	Parameters	Value
GWO (2014)	Convergence constant a N	Decreases Linearly from 2 to 0 20	SMA (2020)	z N	0.03 20
WOA (2017)	Convergence constant a Spiral factor b N	Decreases Linearly from 2 to 0 1 20	GTO (2021)	p $Beta$ w N	0.03 3 8 20
EO (2020)	a_1 a_2 V GP N	2 1 1 0.5 20	AVOA (2021)	Alpha (L_1) Beta (L_2) Gamma (w) P_1 P_2 P_3 N	0.8 0.2 2.5 0.6 0.4 0.6 20
RUN (2021)	a b N	20 12 20	RSA (2022)	Alpha Beta N	0.1 0.1 20
GBO(2020)	pr β_{min} β_{max} N	0.5 0.2 1.2 20	DE	Crossover rate Scaling factor N	0.5 0.5 20
SSA (2017)	$c1$ N	Decreases from 2 to 0 20			

4.2. Sensitivity Analysis of LSO

Extensive experiments have been done to perform a sensitivity analysis of four controlling parameters found in LSO, which are P_e , P_s , P_h , and β . For each parameter of these, extensive experiments have been done using different values for each one to solve two test functions: F57 and F58, and the obtained outcomes, are depicted in Figure 8. This figure shows that the most effective values of these four parameters: P_e , P_s , P_h , and β , for two observed test functions are of 0.9, 0.05, 0.4, and 0.05, respectively.

The first investigated parameter is P_h (responsible for the tradeoff between two boundary checking methods to improve the LSO's searchability), which is analyzed in Figure 8a,b using various randomly-picked values between 0 and 1.0. These figures show that LSO could reach the optimal value for the test function: F58 when $P_h = 0.4$. Based on that, this value is assigned to P_h within the experiments conducted in this study.

For the parameter β (responsible for improving the convergence speed of LSO), Figure 8c,d depicts the performance of LSO under various randomly-picked values between 0 and 0.6 for this parameter over two test problems: F57 and F58. According to this figure, over F57, on one side, the performance of LSO is substantially improved with increasing the value of this parameter even reaching 0.3, and then the performance again deteriorated. On the other side, over F58, LSO has poor performance when increasing the value of this parameter. Therefore, we found that the best value for the parameter β that will be substantially suitable for most test functions is 0.05, since LSO under this value could reach the optimal value for the test problem: F58. It is worth mentioning that this parameter is responsible for accelerating the convergence speed of LSO to reach the near-optimal solution in as low a number of function evaluations as possible. Therefore, an additional experiment has been conducted in this section to depict the convergence speed of LSO under various values for the parameter β over F58 (see Figure 8e). Figure 8e further affirms that the best value for this parameter is 0.05.

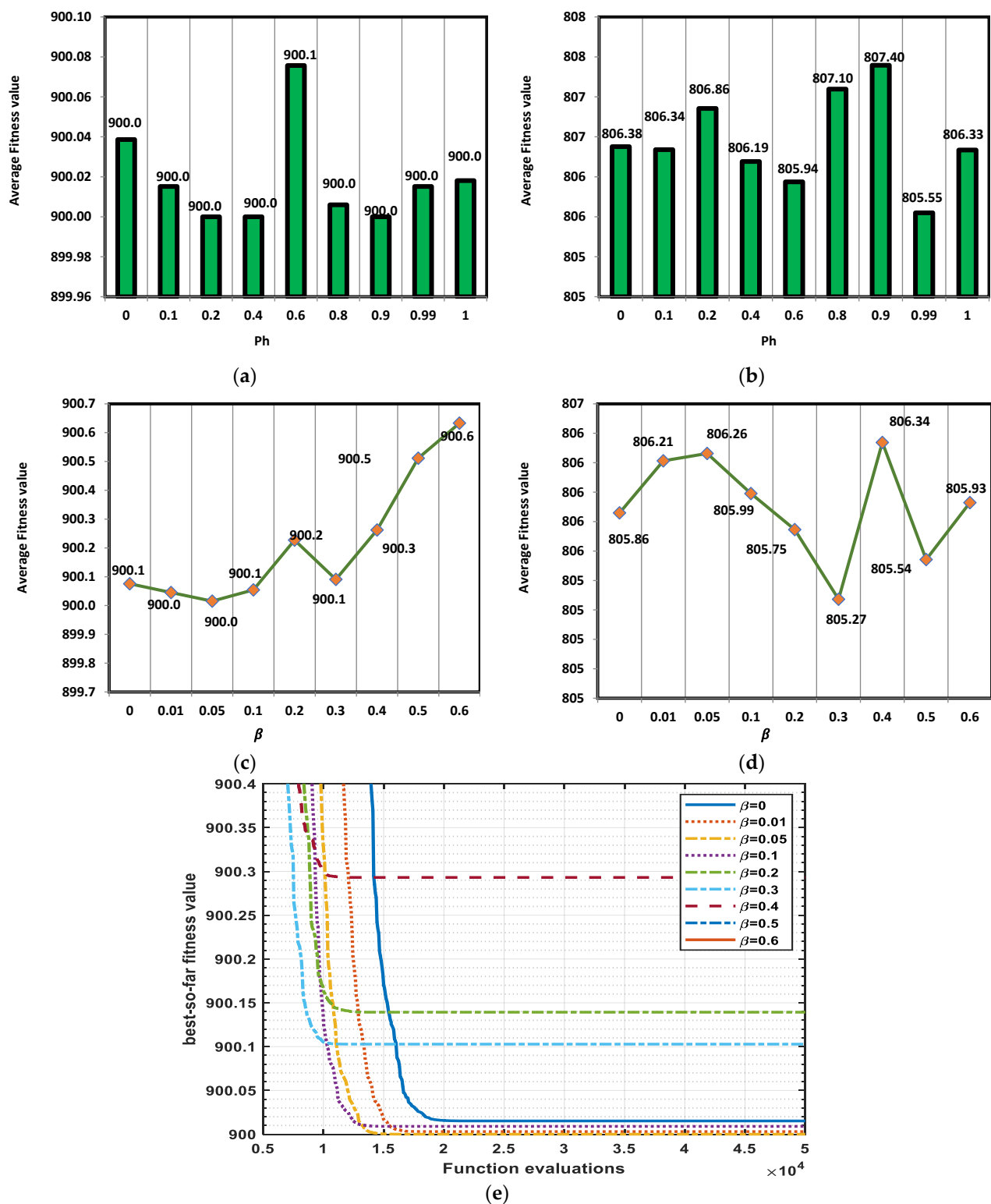


Figure 8. Cont.

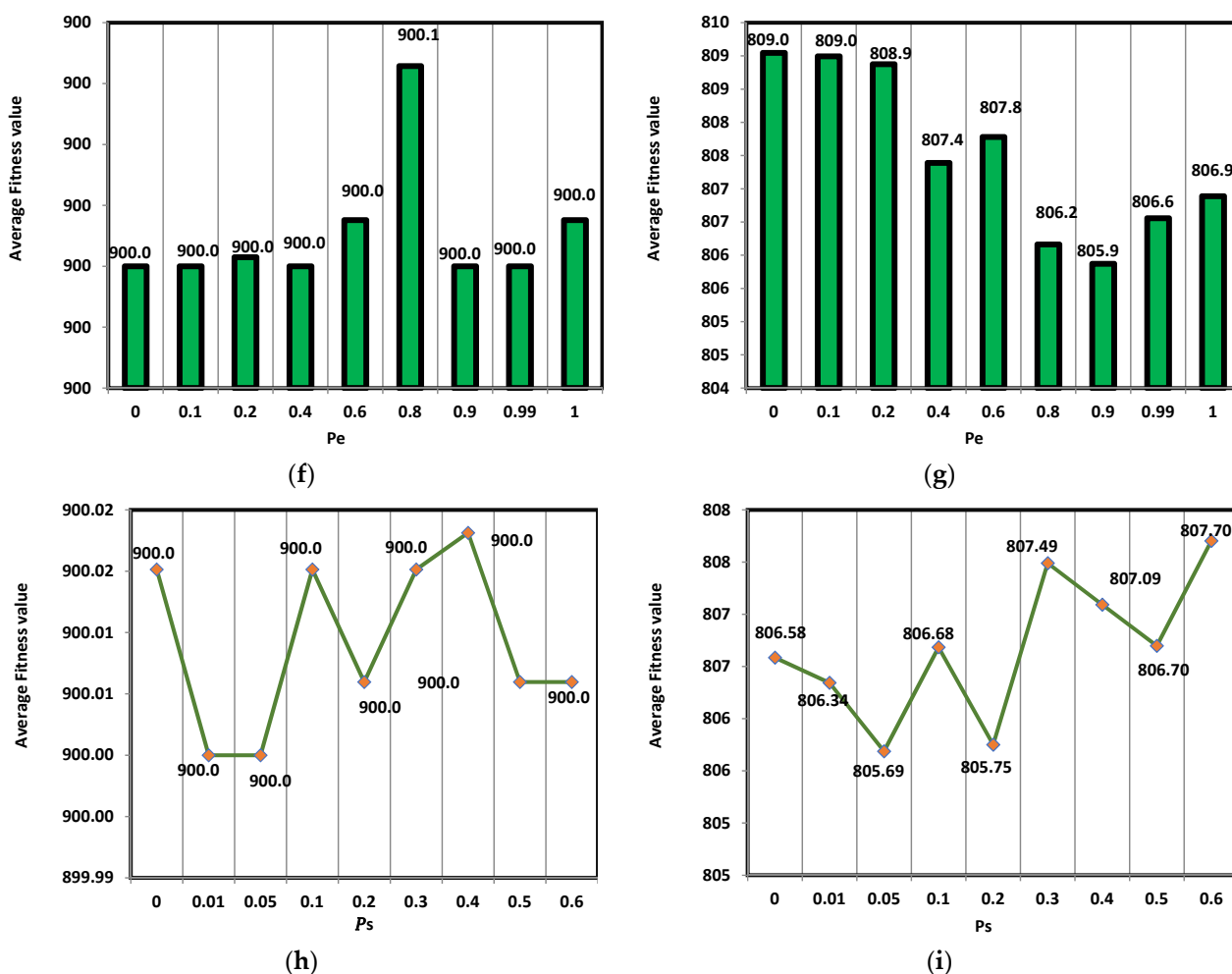


Figure 8. Sensitivity analysis of LSO. (a) Tuning the parameter P_h over F58. (b) Tuning the parameter P_h over F57. (c) Tuning the parameter β over F58. (d) Tuning the parameter β over F57. (e) Tuning the parameter β over F58 in terms of convergence speed. (f) Tuning the parameter P_e over F58. (g) Tuning the parameter P_e over F57. (h) Adjusting the parameter P_s over F58. (i) Adjusting the parameter P_s over F57.

The third parameter is P_e , employed in LSO to exchange between the first and second scattering phases. Figure 8f,g compare the influence of various values for this parameter over the test functions: F57 and F58. According to these figures, the best value for this parameter is 0.9, since LSO under this value could reach 900 and 805.9 for F58 and F57, respectively. Regarding the parameter P_s , which is employed to further promote the first and second scattering stages for improving the exploitation operator of LSO, Figure 8h,i were presented to report the influence of various values for this parameter; these values range between 0 and 0.6. Inspecting these figures shows that LSO reaches the top performance when P_s has a value of 0.05 over two investigated test functions: F57 and F58.

4.3. Evaluation of Exploitation and Exploration Operators

The class of Uni-modal benchmarks has only one global optimal solution. This feature allows testing and validating a metaheuristic's exploitation capabilities. Tables 6 and 7 show that LSO is competitive with some other comparators for F1, F2, and F3. For F3 and F5, LSO has inferior performance compared to some of the recently-published rival algorithms. In general, LSO proves that it has a competitive exploitation operator. Multi-modal classes can efficiently discover the exploration of metaheuristics, as they have many optimal solutions. As observed in Table 6, LSO is able to reach the optimal solution for 13 benchmarks,

especially fixed-dimension ones, including F11–F20. In addition, LSO is competitive with the other comparators for F6–F8. To affirm the difference between the outcomes produced by LSO and those of the rival algorithms, the Wilcoxon rank-sum test is employed to compute the p -values, which determine that there is a difference when its value is less than 5%; otherwise, there is no difference. Table 7 introduces the p -values on unimodal and multimodal test functions between LSO and each rival algorithm. These values clarify that there are differences between the outcomes of LSO and most rival algorithms on most test functions. NaN in this table indicates that the independent outcomes of LSO and the corresponding optimizer are the same. As a result, the results and discussion is given herein assure the prosperity of LSO's exploration and exploitation capabilities.

4.4. LSO for Challengeable CEC2014

Additional validation is carried out on the CEC-2014 test suite in order to ensure that the proposed and other methods perform in accordance with expectations. With the help of this collection of test functions, you can determine whether or not an algorithm has the ability to explore, escape from local minima, and exploit. The test functions are divided into four categories: unimodal, multimodal, hybrid, and composition. The test suite's characteristics are described in greater detail in Appendix A Tables A1–A7. The dimensions taken into consideration within our experiments are all ten. Table 8 shows the average and standard deviation values and the rank metric obtained by rival optimizers and others on CEC2014. Inspecting this table demonstrates that LSO could rank first for 23 out of 30 test functions, while its performance for the other test functions is competitive with some of the other optimizers. The average of rank values presented in Table 8 for each test function to show the best-performance order of the algorithms is computed and displayed in Figure 9. This figure shows the superior performance of LSO because it could come in the first rank with a value of 1.7, followed by DE with a value of 4.4, while RSA is the worst performing one.

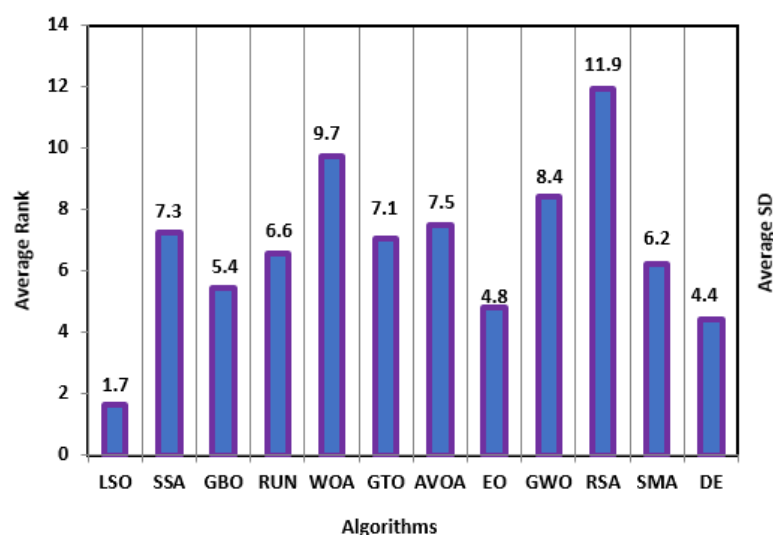


Figure 9. Average rank of each optimizer on all CEC2014.

Table 6. Avr and SD of CEC2005 for 25 independent runs (The bolded value is the best overall).

F	Index	LSO	SSA	GBO	RUN	WOA	GTO	AVOA	EO	GWO	RSA	SMA	DE
Unimodal													
F1	Avr	0.00×10^0	4.27×10^{-7}	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0	2.5×10^{-145}	3.90×10^{-70}	0.00×10^0	0.00×10^0	3.20×10^{-3}
	SD	0.00×10^0	7.68×10^{-8}	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0	7.4×10^{-145}	7.34×10^{-70}	0.00×10^0	0.00×10^0	1.75×10^{-2}
	Rank	1.00×10^0	1.10×10^1	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	9.00×10^0	1.00×10^1	1.00×10^0	1.00×10^0	1.20×10^1
F2	Avr	0.00×10^0	1.94×10^1	1.6×10^{-283}	1.5×10^{-238}	0.00×10^0	0.00×10^0	0.00×10^0	3.58×10^{-84}	1.33×10^{-41}	0.00×10^0	0.00×10^0	3.70×10^{-5}
	SD	0.00×10^0	1.20×10^1	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0	1.02×10^{-83}	1.04×10^{-41}	0.00×10^0	0.00×10^0	1.57×10^{-5}
	Rank	1.00×10^0	1.30×10^1	7.00×10^0	8.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	9.00×10^0	1.00×10^1	1.00×10^0	1.00×10^0	1.10×10^1
F3	Avr	0.00×10^0	2.92×10^4	0.00×10^0	0.00×10^0	7.25×10^5	0.00×10^0	0.00×10^0	2.63×10^{-10}	8.42×10^{-4}	0.00×10^0	0.00×10^0	3.44×10^5
	SD	0.00×10^0	1.24×10^4	0.00×10^0	0.00×10^0	1.42×10^5	0.00×10^0	0.00×10^0	1.01×10^{-9}	3.97×10^{-3}	0.00×10^0	0.00×10^0	3.81×10^4
	Rank	1.00×10^0	1.00×10^1	1.00×10^0	1.00×10^0	1.30×10^1	1.00×10^0	1.00×10^0	7.00×10^0	8.00×10^0	1.00×10^0	1.00×10^0	1.20×10^1
F4	Avr	0.00×10^0	3.22×10^1	3.9×10^{-253}	1.3×10^{-196}	7.78×10^1	0.00×10^0	0.00×10^0	2.87×10^{-16}	2.72×10^{-8}	0.00×10^0	0.00×10^0	9.59×10^1
	SD	0.00×10^0	2.81×10^0	0.00×10^0	0.00×10^0	2.30×10^1	0.00×10^0	0.00×10^0	1.57×10^{-15}	1.26×10^{-7}	0.00×10^0	0.00×10^0	1.17×10^0
	Rank	1.00×10^0	1.00×10^1	6.00×10^0	7.00×10^0	1.10×10^1	1.00×10^0	1.00×10^0	8.00×10^0	9.00×10^0	1.00×10^0	1.00×10^0	1.30×10^1
F5	Avr	9.70×10^1	3.85×10^2	9.31×10^1	9.65×10^1	9.74×10^1	8.30×10^{-3}	1.03×10^{-5}	9.43×10^1	9.76×10^1	9.90×10^1	1.88×10^0	1.66×10^3
	SD	6.18×10^{-1}	4.25×10^2	2.49×10^0	1.12×10^0	6.96×10^{-1}	1.14×10^{-2}	1.35×10^{-5}	9.49×10^{-1}	6.38×10^{-1}	0.00×10^0	1.87×10^0	5.76×10^3
	Rank	7.00×10^0	1.10×10^1	4.00×10^0	6.00×10^0	8.00×10^0	2.00×10^0	1.00×10^0	5.00×10^0	9.00×10^0	1.00×10^1	3.00×10^0	1.20×10^1
High-dimensional multimodal													
F6	Avr	0.00×10^0	2.33×10^2	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0	7.57×10^2
	SD	0.00×10^0	5.60×10^1	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0	3.26×10^1
	Rank	1.00×10^0	1.20×10^1	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^1	1.30×10^1
F7	Avr	8.88×10^{-16}	5.71×10^0	8.88×10^{-16}	8.88×10^{-16}	3.14×10^{-15}	8.88×10^{-16}	8.88×10^{-16}	5.63×10^{-15}	2.61×10^{-14}	8.88×10^{-16}	8.88×10^{-16}	2.23×10^{-4}
	SD	0.00×10^0	1.30×10^0	0.00×10^0	0.00×10^0	2.72×10^{-15}	0.00×10^0	0.00×10^0	1.70×10^{-15}	3.41×10^{-15}	0.00×10^0	0.00×10^0	3.05×10^{-4}
	Rank	1.00×10^0	6.00×10^0	1.00×10^0	1.00×10^0	2.00×10^0	1.00×10^0	1.00×10^0	3.00×10^0	5.00×10^0	1.00×10^0	1.00×10^0	4.00×10^0
F8	Avr	0.00×10^0	9.68×10^{-3}	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0	6.37×10^{-4}	0.00×10^0	0.00×10^0	3.31×10^{-4}
	SD	0.00×10^0	5.56×10^{-3}	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0	3.49×10^{-3}	0.00×10^0	0.00×10^0	1.80×10^{-3}
	Rank	1.00×10^0	4.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	3.00×10^0	1.00×10^0	1.00×10^0	2.00×10^0
F9	Avr	4.67×10^{-3}	1.52×10^1	1.25×10^{-6}	1.95×10^{-6}	9.70×10^{-3}	8.72×10^{-7}	5.07×10^{-9}	1.85×10^{-3}	2.91×10^{-1}	1.33×10^0	6.83×10^{-4}	1.59×10^3
	SD	1.91×10^{-3}	2.95×10^0	1.11×10^{-6}	6.52×10^{-6}	3.24×10^{-3}	1.14×10^{-6}	2.63×10^{-9}	1.79×10^{-3}	7.27×10^{-2}	0.00×10^0	9.08×10^{-4}	6.68×10^3
	Rank	7.00×10^0	1.10×10^1	3.00×10^0	4.00×10^0	8.00×10^0	2.00×10^0	1.00×10^0	6.00×10^0	9.00×10^0	1.00×10^1	5.00×10^0	1.20×10^1
F10	Avr	8.42×10^0	1.66×10^2	1.87×10^0	1.11×10^{-1}	1.50×10^0	3.95×10^{-4}	3.05×10^{-9}	3.70×10^0	6.55×10^0	9.72×10^0	5.29×10^{-3}	2.54×10^4
	SD	1.47×10^0	1.71×10^1	2.63×10^0	1.15×10^{-1}	5.79×10^{-1}	2.03×10^{-3}	4.10×10^{-9}	1.05×10^0	4.13×10^{-1}	8.77×10^{-1}	5.86×10^{-3}	6.44×10^4
	Rank	9.00×10^0	1.10×10^1	6.00×10^0	4.00×10^0	5.00×10^0	2.00×10^0	1.00×10^0	7.00×10^0	8.00×10^0	1.00×10^1	3.00×10^0	1.20×10^1
Fixed-dimensional multimodal													
F11	Avr	9.98×10^{-1}	9.98×10^{-1}	9.98×10^{-1}	4.63×10^0	2.24×10^0	9.98×10^{-1}	1.06×10^0	1.06×10^0	5.76×10^0	4.47×10^0	9.98×10^{-1}	1.23×10^0
	SD	0.00×10^0	2.24×10^{-16}	4.12×10^{-17}	3.84×10^0	2.49×10^0	0.00×10^0	3.62×10^{-1}	3.62×10^{-1}	4.64×10^0	3.30×10^0	8.55×10^{-15}	9.59×10^{-1}
	Rank	1.00×10^0	1.00×10^0	1.00×10^0	1.20×10^1	5.00×10^0	1.00×10^0	2.00×10^0	3.00×10^0	7.00×10^0	6.00×10^0	1.00×10^0	4.00×10^0

Table 6. Cont.

F	Index	LSO	SSA	GBO	RUN	WOA	GTO	AVOA	EO	GWO	RSA	SMA	DE
F12	Avr	3.07×10^{-4}	8.50×10^{-4}	4.30×10^{-4}	7.04×10^{-4}	6.30×10^{-4}	5.52×10^{-4}	3.08×10^{-4}	4.38×10^{-3}	6.32×10^{-3}	1.58×10^{-3}	4.82×10^{-4}	2.68×10^{-3}
	SD	2.01×10^{-19}	2.64×10^{-4}	3.17×10^{-4}	4.62×10^{-4}	3.47×10^{-4}	4.12×10^{-4}	4.17×10^{-8}	8.13×10^{-3}	9.35×10^{-3}	1.03×10^{-3}	2.95×10^{-4}	6.01×10^{-3}
	Rank	1.00×10^0	9.00×10^0	3.00×10^0	7.00×10^0	6.00×10^0	5.00×10^0	1.00×10^0	1.20×10^1	1.30×10^1	2.00×10^1	4.00×10^0	1.10×10^1
F13	Avr	-1.03×10^0	-1.03×10^0	-1.03×10^0	-1.03×10^0	-1.03×10^0	-1.03×10^0	-1.03×10^0	-1.03×10^0	-1.03×10^0	-1.03×10^0	-1.03×10^0	-1.03×10^0
	SD	6.78×10^{-16}	3.72×10^{-15}	6.78×10^{-16}	2.05×10^{-13}	5.36×10^{-12}	6.52×10^{-16}	5.13×10^{-16}	6.39×10^{-16}	8.86×10^{-10}	1.59×10^{-3}	1.72×10^{-11}	6.78×10^{-16}
	Rank	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0
F14	Avr	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	4.18×10^{-1}	3.98×10^{-1}	3.98×10^{-1}
	SD	0.00×10^0	2.70×10^{-15}	0.00×10^0	1.17×10^{-11}	3.58×10^{-7}	0.00×10^0	0.00×10^0	0.00×10^0	5.93×10^{-8}	4.66×10^{-2}	3.82×10^{-9}	0.00×10^0
	Rank	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.30×10^1	1.00×10^0	1.00×10^0
F15	Avr	3.00×10^0	3.00×10^0	3.00×10^0	3.00×10^0	3.00×10^0	3.00×10^0	3.00×10^0	3.00×10^0	3.00×10^0	3.00×10^0	3.00×10^0	3.00×10^0
	SD	1.31×10^{-15}	2.94×10^{-14}	1.40×10^{-15}	1.43×10^{-13}	1.63×10^{-5}	1.33×10^{-15}	5.67×10^{-8}	1.39×10^{-15}	3.76×10^{-6}	5.55×10^{-5}	2.27×10^{-13}	1.83×10^{-15}
	Rank	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0
F16	Avr	-3.86×10^0	-3.86×10^0	-3.86×10^0	-3.86×10^0	-3.86×10^0	-3.86×10^0	-3.86×10^0	-3.86×10^0	-3.86×10^0	-3.80×10^0	-3.86×10^0	-3.86×10^0
	SD	2.71×10^{-15}	5.11×10^{-15}	2.71×10^{-15}	4.64×10^{-6}	2.98×10^{-3}	2.59×10^{-15}	2.20×10^{-15}	2.00×10^{-3}	2.79×10^{-3}	5.08×10^{-2}	3.64×10^{-9}	2.71×10^{-15}
	Rank	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.00×10^0	1.30×10^1	1.00×10^0	1.00×10^0
F17	Avr	-3.32×10^0	-3.07×10^0	-3.27×10^0	-3.24×10^0	-2.71×10^0	-3.07×10^0	-3.07×10^0	-3.24×10^0	-3.19×10^0	-1.74×10^0	-3.03×10^0	-3.12×10^0
	SD	1.32×10^{-15}	1.57×10^{-1}	6.62×10^{-2}	7.38×10^{-2}	4.14×10^{-1}	2.49×10^{-1}	3.31×10^{-1}	8.05×10^{-2}	1.19×10^{-1}	5.78×10^{-1}	3.05×10^{-1}	1.01×10^{-1}
	Rank	1.00×10^0	9.00×10^0	2.00×10^0	4.00×10^0	1.10×10^1	7.00×10^0	8.00×10^0	3.00×10^0	5.00×10^0	1.30×10^1	1.00×10^1	6.00×10^0
F18	Avr	-1.02×10^1	-5.12×10^0	-7.36×10^0	-6.31×10^0	-4.51×10^0	-7.44×10^0	-7.25×10^0	-7.18×10^0	-7.36×10^0	-4.79×10^0	-6.37×10^0	-3.64×10^0
	SD	7.23×10^{-15}	3.16×10^0	2.44×10^0	2.47×10^0	2.33×10^0	2.48×10^0	2.84×10^0	3.20×10^0	3.50×10^0	9.60×10^{-1}	2.53×10^0	2.28×10^0
	Rank	1.00×10^0	9.00×10^0	4.00×10^0	8.00×10^0	1.10×10^1	2.00×10^0	5.00×10^0	6.00×10^0	3.00×10^0	1.00×10^1	7.00×10^0	1.20×10^1
F19	Avr	-1.04×10^1	-9.27×10^0	-8.63×10^0	-1.04×10^1	-9.38×10^0	-1.04×10^1	-1.04×10^1	-1.00×10^1	-1.00×10^1	-5.09×10^0	-1.04×10^1	-9.35×10^0
	SD	1.48×10^{-15}	2.35×10^0	2.55×10^0	6.54×10^{-10}	2.36×10^0	7.38×10^{-16}	1.14×10^{-15}	1.34×10^0	1.35×10^0	8.85×10^{-7}	1.91×10^{-5}	2.42×10^0
	Rank	1.00×10^0	6.00×10^0	7.00×10^0	1.00×10^0	4.00×10^0	1.00×10^0	1.00×10^0	2.00×10^0	3.00×10^0	8.00×10^0	1.00×10^0	5.00×10^0
F20	Avr	-1.05×10^1	-9.74×10^0	-8.91×10^0	-1.04×10^1	-9.23×10^0	-1.05×10^1	-1.05×10^1	-1.00×10^1	-1.04×10^1	-5.13×10^0	-1.05×10^1	-9.16×10^0
	SD	1.81×10^{-15}	2.10×10^0	2.52×10^0	9.87×10^{-1}	2.42×10^0	2.06×10^{-15}	3.32×10^{-15}	1.65×10^0	9.87×10^{-1}	1.85×10^{-6}	1.91×10^{-5}	2.80×10^0
	Rank	1.00×10^0	5.00×10^0	8.00×10^0	2.00×10^0	6.00×10^0	1.00×10^0	1.00×10^0	4.00×10^0	3.00×10^0	9.00×10^0	1.00×10^0	7.00×10^0

Table 7. *p*-values of LSO with each rival optimizer on CEC2005 test suite (F1–F20).

Fun	SCA	SSA	GBO	RUN	WOA	GTO	AVOA	EO	GWO	RFO	SMA
Unimodal											
F1	1.21×10^{-12}	NaN	NaN	NaN	NaN	NaN	1.21×10^{-12}	1.21×10^{-12}	NaN	NaN	1.21×10^{-12}
F2	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	NaN	NaN	NaN	1.21×10^{-12}	1.21×10^{-12}	NaN	NaN	1.21×10^{-12}
F3	3.75×10^{-11}	5.37×10^{-6}	5.37×10^{-6}	2.26×10^{-11}	5.37×10^{-6}	5.37×10^{-6}	6.61×10^{-1}	6.61×10^{-1}	5.37×10^{-6}	5.37×10^{-6}	2.26×10^{-11}
F4	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	NaN	NaN	1.21×10^{-12}	1.21×10^{-12}	NaN	NaN	1.21×10^{-12}
F5	1.56×10^{-8}	5.53×10^{-8}	4.51×10^{-2}	6.97×10^{-3}	3.02×10^{-11}	3.02×10^{-11}	2.44×10^{-9}	1.17×10^{-4}	1.21×10^{-12}	3.02×10^{-11}	3.02×10^{-11}
High-dimensional multimodal											
F6	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	5.69×10^{-1}	3.02×10^{-11}	3.02×10^{-11}	3.47×10^{-10}	3.02×10^{-11}	1.21×10^{-12}	3.02×10^{-11}	5.57×10^{-10}
F7	3.02×10^{-11}	1.11×10^{-3}	1.56×10^{-2}	3.52×10^{-7}	1.38×10^{-2}	6.79×10^{-2}	2.87×10^{-10}	4.50×10^{-11}	5.32×10^{-3}	8.88×10^{-1}	3.02×10^{-11}
F8	4.64×10^{-3}	6.74×10^{-6}	1.89×10^{-4}	8.99×10^{-11}	3.02×10^{-11}	3.69×10^{-11}	6.05×10^{-7}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}
F9	1.21×10^{-12}	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1.21×10^{-12}
F10	1.21×10^{-12}	NaN	NaN	2.75×10^{-5}	NaN	NaN	3.37×10^{-13}	7.73×10^{-13}	NaN	NaN	1.21×10^{-12}
Fixed-dimensional multimodal											
F11	1.21×10^{-12}	NaN	NaN	NaN	NaN	NaN	NaN	3.34×10^{-1}	NaN	NaN	1.21×10^{-12}
F12	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	5.00×10^{-9}	3.02×10^{-11}	3.02×10^{-11}	8.20×10^{-7}	3.02×10^{-11}	1.21×10^{-12}	1.46×10^{-10}	1.21×10^{-10}
F13	3.02×10^{-11}	9.92×10^{-11}	3.02×10^{-11}	1.33×10^{-10}	3.02×10^{-11}	3.02×10^{-11}	5.57×10^{-10}	8.10×10^{-10}	1.21×10^{-8}	3.02×10^{-11}	3.15×10^{-2}
F14	3.00×10^{-13}	3.34×10^{-1}	1.20×10^{-12}	1.21×10^{-12}	NaN	2.52×10^{-7}	4.19×10^{-2}	1.21×10^{-12}	1.21×10^{-12}	1.20×10^{-12}	1.61×10^{-1}
F15	1.58×10^{-9}	5.33×10^{-6}	7.55×10^{-10}	3.28×10^{-9}	3.43×10^{-8}	7.97×10^{-9}	1.89×10^{-9}	1.58×10^{-9}	3.55×10^{-10}	4.69×10^{-9}	8.23×10^{-5}
F16	1.19×10^{-12}	NaN	4.57×10^{-12}	1.21×10^{-12}	4.18×10^{-2}	1.47×10^{-9}	1.09×10^{-2}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	NaN
F17	4.15×10^{-8}	NaN	1.66×10^{-11}	4.57×10^{-12}	NaN	NaN	NaN	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	NaN
F18	6.43×10^{-12}	1.70×10^{-1}	6.46×10^{-12}	6.46×10^{-12}	3.11×10^{-1}	6.46×10^{-12}	8.24×10^{-2}	6.46×10^{-12}	6.46×10^{-12}	6.46×10^{-12}	4.30×10^{-1}
F19	1.20×10^{-12}	NaN	1.21×10^{-12}	1.21×10^{-12}	5.54×10^{-3}	3.67×10^{-10}	8.15×10^{-2}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	NaN
F20	8.44×10^{-12}	1.62×10^{-4}	1.99×10^{-11}	1.99×10^{-11}	1.15×10^{-6}	4.90×10^{-11}	6.46×10^{-6}	2.21×10^{-11}	3.16×10^{-12}	8.44×10^{-12}	1.88×10^{-5}

In terms of the standard deviation, Figure 10 displays the average SD values of 25 independent runs for each CEC-2014 test function; this figure discloses that the outcomes obtained by LSO within these independent runs are substantially similar since it could reach less average SD of 23, while RSA has the worst average SD. Finally, the Wilcoxon rank-sum statistical test is employed to show the difference between the outcomes of LSO and each rival algorithm; this test relies on two hypotheses: the null hypothesis, indicating that there is no difference, and the alternative one, indicating that there is difference between the outcomes of each pair of algorithms. This test determines the accepted hypothesis based on the confidence level and p -value returned after comparing the outcomes of each pair of algorithms. Within our experiments, the confidence level is 0.05. The obtained p -value between LSO and each rival algorithm is presented in Table 9. The majority of p -values presented in this table is less than 5%, which notifies us that the alternative hypothesis is accepted; hence, the outcome of LSO is different from those of the other compared algorithms.

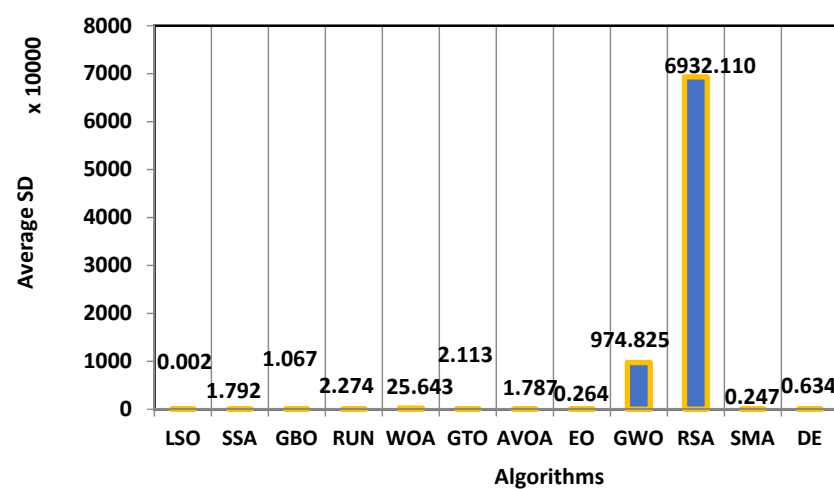


Figure 10. Average SD of each optimizer on all CEC2014.

4.5. LSO for Challengeable CEC2017

This section compares the performance of LSO and other optimizers using the CEC2017 test suite to further validate the performance of LSO against the comparators for more challenging mathematical test functions [126]. CEC2017 is composed of four mathematical function families: unimodal (F51–F52), multimodal (F53–F59), composition (F60–F69), and hybrid (F70–F79). As previously described, unimodal test functions are preferable for evaluating the exploitation operator of optimization algorithms because they involve only one global best solution, and multimodal test functions contain multiple local optimal solutions, which makes them particularly well-suited for evaluating the exploration operator of newly proposed optimizers; while composition and hybrid test functions have been designed to evaluate the optimization algorithms' ability to escape out of local optima. The dimension of this benchmark is set to 10 within the conducted experiments in this section. Appendix A Tables A1–A7 contains the characteristics of the CEC2017 benchmark.

Table 10 shows the Avr, SD, and Rank values of 25 independent findings obtained by this suite's proposed and rival optimizers. According to this table, LSO comes in the first rank compared to all optimizers for unimodal, multimodal, composition and hybrid test functions since it could reach better Avr and SD for all test functions. Figures 11 and 12 display the average of the rank and standard deviation values presented in Table 10 for all test functions for each algorithm. According to these figures, LSO is the best since it occupies the 1st rank with a value of 1 and has the lowest standard deviation of 32, while RSA is the worst.

Table 8. Avr and SD of CEC2014 for 25 independent runs (The bolded value is the best overall).

F	Index	LSO	SSA	GBO	RUN	WOA	GTO	AVOA	EO	GWO	RSA	SMA	DE
Unimodal													
F21	Avr	1.00×10^2	5.29×10^5	1.73×10^3	1.29×10^5	8.21×10^6	2.81×10^3	1.87×10^5	5.06×10^4	5.03×10^6	1.32×10^8	1.53×10^5	3.05×10^3
	SD	7.09×10^{-11}	5.03×10^5	2.04×10^3	1.11×10^5	6.51×10^6	3.88×10^3	1.12×10^5	6.63×10^4	3.43×10^6	8.55×10^7	4.14×10^4	7.78×10^3
	Rank	1.00	9.00	2.00	6.00	12.00	3.00	8.00	5.00	10.00	13.00	7.00	4.00
F22	Avr	200.00	3802.99	272.45	4382.91	940,880.88	557.78	3404.48	1.30×10^3	5.69×10^7	7.09×10^9	5.94×10^3	7.20×10^2
	SD	4.22×10^{-14}	3.65×10^3	1.30×10^2	5.25×10^3	7.23×10^5	8.18×10^2	3.67×10^3	1.65×10^3	2.88×10^8	1.99×10^9	3.99×10^3	2.07×10^3
	Rank	1.00	7.00	2.00	8.00	10.00	3.00	6.00	5.00	11.00	13.00	9.00	4.00
F23	Avr	300.00	3450.21	314.70	1401.82	43,769.75	315.07	1571.60	426.51	7388.05	8368.14	2407.10	301.79
	SD	0.00×10^0	2845.55	21.18	566.37	28,043.51	50.45	1128.64	203.06	4648.72	2959.58	1895.90	9.80
	Rank	1.00	9.00	3.00	6.00	13.00	4.00	7.00	5.00	11.00	12.00	8.00	2.00
Multimodal													
F24	Avr	405.36	425.87	424.51	416.08	438.61	419.47	419.49	423.67	434.97	1735.39	424.29	425.97
	SD	11.85	15.12	15.98	15.73	22.17	16.71	17.93	16.02	18.96	992.19	15.13	14.33
	Rank	1.00	8.00	7.00	2.00	11.00	3.00	4.00	5.00	10.00	13.00	6.00	9.00
F25	Avr	520.03	520.04	520.07	520.09	520.14	520.11	520.06	520.11	520.42	520.43	519.43	520.21
	SD	1.18×10^{-2}	8.81×10^{-2}	9.18×10^{-2}	1.06×10^{-1}	9.39×10^{-2}	7.89×10^{-2}	7.39×10^{-2}	7.25×10^{-2}	0.08	0.08	3.66	0.06
	Rank	2.00	3.00	5.00	6.00	9.00	7.00	4.00	8.00	12.00	13.00	1.00	10.00
F26	Avr	600.32	603.66	604.32	605.49	607.99	605.26	605.99	601.51	6.02×10^2	609.69	604.19	600.84
	SD	0.61	1.59	1.65	1.14	1.76	1.53	1.48	1.21	1.16×10^0	0.80	1.24	0.83
	Rank	1.00	5.00	7.00	9.00	12.00	8.00	10.00	3.00	4.00×10^0	13.00	6.00	2.00
F27	Avr	700.02	700.26	700.17	700.35	701.09	700.29	700.41	700.05	701.20	795.27	700.24	700.08
	SD	0.01	0.10	0.08	0.23	0.46	0.19	0.31	0.04	0.95	27.14	0.12	0.10
	Rank	1.00	6.00	4.00	8.00	10.00	7.00	9.00	2.00	11.00	13.00	5.00	3.00
F28	Avr	800.50	821.33	821.40	821.03	844.05	823.18	813.43	807.33	812.15	876.28	801.79	800.66
	SD	6.27×10^{-1}	1.11×10^1	9.72×10^0	7.53×10^0	1.58×10^1	1.02×10^1	5.44×10^0	4.71×10^0	5.70	8.27	1.15	1.05
	Rank	1.00	8.00	9.00	7.00	12.00	10.00	6.00	4.00	5.00	13.00	3.00	2.00
F29	Avr	907.18	920.93	925.22	937.34	950.57	931.37	931.48	914.54	914.91	960.42	915.39	914.65
	SD	2.58	9.25	8.72	5.75	19.54	10.36	9.98	5.91	6.21	5.63	5.69	3.48
	Rank	1.00	6.00	7.00	10.00	12.00	8.00	9.00	2.00	4.00	13.00	5.00	3.00
F30	Avr	1010.54	1571.44	1457.97	1186.07	1610.08	1580.21	1104.67	1169.31	1370.39	2107.30	1145.39	1068.08
	SD	30.00	257.58	238.56	120.79	352.70	273.00	76.89	146.30	141.26	167.86	110.93	53.37
	Rank	1.00	9.00	8.00	6.00	11.00	10.00	3.00	5.00	7.00	13.00	4.00	2.00
F31	Avr	1888.08	2.37×10^3	1.98×10^3	1.69×10^3	2.99×10^3	1.99×10^3	1.87×10^3	1.71×10^3	1823.92	2501.55	1950.17	2447.27
	SD	202.29	3.49×10^2	3.28×10^2	2.35×10^2	5.38×10^2	3.30×10^2	3.50×10^2	3.17×10^2	405.02	203.72	258.79	193.51
	Rank	5.00	9.00	7.00	1.00	13.00	8.00	4.00	2.00	3.00	11.00	6.00	10.00

Table 8. Cont.

F	Index	LSO	SSA	GBO	RUN	WOA	GTO	AVOA	EO	GWO	RSA	SMA	DE
F43	Avr	2581.99	2629.46	2500.00	2500.00	2616.13	2500.00	2500.00	2629.46	2634.66	2500.00	2500.00	2629.46
	SD	63.45	0.00	0.00	0.00	46.43	0.00	0.00	0.00	4.52	0.00	0.00	0.00
	Rank	7.00	10.00	1.00	2.00	8.00	3.00	4.00	9.00	12.00	5.00	6.00	11.00
F44	Avr	2515.12	2552.08	2532.18	2547.45	2574.63	2582.52	2582.07	2580.26	2543.12	2538.92	2599.43	2536.95
	SD	3.96	8.58	9.38	25.72	30.06	23.13	28.47	27.82	36.12	29.87	1.78	29.67
	Rank	1.00	8.00	3.00	7.00	9.00	12.00	11.00	10.00	6.00	5.00	13.00	4.00
F45	Avr	2633.50	2696.49	2678.16	2682.90	2696.47	2694.01	2696.55	2696.56	2690.04	2693.41	2699.88	2694.51
	SD	10.38	1.17×10^1	3.10×10^1	2.32×10^1	1.05×10^1	1.29×10^1	1.36×10^1	8.76×10^0	21.16	18.89	0.65	17.36
	Rank	1.00	1.00×10^1	2.00×10^0	4.00×10^0	9.00×10^0	7.00×10^0	1.10×10^1	1.20×10^1	5.00	6.00	13.00	8.00
F46	Avr	2700.14	2700.65	2700.18	2700.28	2700.17	2700.34	2700.24	2700.30	2703.40	2703.48	2709.86	2700.22
	SD	0.04	0.12	0.08	0.17	0.08	0.17	0.12	0.15	18.25	18.23	19.72	0.08
	Rank	1.00	10.00	3.00	7.00	2.00	9.00	6.00	8.00	11.00	12.00	13.00	5.00
F47	Avr	2823.87	3002.57	3010.43	2842.06	2874.38	3084.23	2854.69	2893.57	3027.04	3032.32	2900.00	2893.39
	SD	154.31	171.71	158.02	90.03	66.44	135.58	83.53	35.20	116.70	98.70	0.00	36.19
	Rank	1.00	9.00	10.00	2.00	4.00	13.00	3.00	6.00	11.00	12.00	7.00	5.00
F48	Avr	3155.87	3278.03	3209.81	3000.00	3000.00	3349.79	3000.00	3000.00	3255.21	3248.91	3000.00	3000.00
	SD	67.80	53.44	58.10	0.00	0.00	159.11	0.00	0.00	79.92	78.45	0.00	0.00
	Rank	7.00	12.00	9.00	1.00	2.00	13.00	3.00	4.00	11.00	10.00	5.00	6.00
F49	Avr	3076.80	9000.02	61,591.92	183,714.47	3824.55	244,952.31	74,649.01	3643.30	413,389.22	545,401.12	3100.00	36,135.91
	SD	48.99	6617.79	314,631.79	551,709.97	588.88	625,545.37	390,869.63	622.98	836,534.22	1.02×10^6	0.00	179,113.96
	Rank	1.00	5.00	7.00	10.00	4.00	11.00	8.00	3.00	12.00	1.30×10^1	2.00	6.00
F50	Avr	3523.72	4608.66	4241.32	3990.04	4470.61	5031.79	3991.63	4425.03	3810.93	4322.67	3200.00	3608.58
	SD	52.44	608.68	508.03	545.97	451.97	982.39	332.96	523.84	304.68	692.91	0.00	307.80
	Rank	3.00	12.00	8.00	6.00	11.00	13.00	7.00	10.00	5.00	9.00	1.00	4.00

Table 9. *p*-values of LSO with each rival optimizer on CEC-2014 test suite (F21–F50).

Fun	SSA	GBO	RUN	WOA	GTO	AVOA	EO	GWO	RSA	SMA	DE
Unimodal											
F21	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}
F22	5.20×10^{-12}	5.20×10^{-12}	5.20×10^{-12}	5.20×10^{-12}	5.20×10^{-12}	5.20×10^{-12}	5.20×10^{-12}	5.20×10^{-12}	5.20×10^{-12}	5.20×10^{-12}	7.85×10^{-1}
F23	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	2.16×10^{-2}
Multimodal											
F24	4.34×10^{-11}	4.49×10^{-8}	9.41×10^{-8}	1.41×10^{-9}	1.52×10^{-7}	1.20×10^{-7}	2.61×10^{-8}	3.42×10^{-10}	1.93×10^{-11}	3.86×10^{-9}	2.36×10^{-8}
F25	5.57×10^{-10}	3.79×10^{-1}	8.29×10^{-6}	1.85×10^{-8}	2.13×10^{-4}	8.77×10^{-1}	8.48×10^{-9}	3.02×10^{-11}	3.02×10^{-11}	1.43×10^{-5}	3.02×10^{-11}
F26	3.02×10^{-11}	6.07×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.69×10^{-11}	3.34×10^{-11}	1.19×10^{-6}	1.41×10^{-9}	3.02×10^{-11}	4.50×10^{-11}	1.90×10^{-1}
F27	3.02×10^{-11}	1.21×10^{-10}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	1.78×10^{-4}	3.02×10^{-11}	3.02×10^{-11}	4.08×10^{-11}	2.84×10^{-1}
F28	1.55×10^{-11}	1.54×10^{-11}	1.55×10^{-11}	1.55×10^{-11}	1.54×10^{-11}	1.54×10^{-11}	5.78×10^{-11}	1.55×10^{-11}	1.55×10^{-11}	4.68×10^{-9}	7.08×10^{-1}
F29	3.02×10^{-11}	6.06×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	1.96×10^{-10}	4.50×10^{-11}	1.73×10^{-6}	1.25×10^{-7}	3.02×10^{-11}	1.10×10^{-8}	1.29×10^{-9}
F30	2.99×10^{-11}	2.35×10^{-10}	4.45×10^{-11}	5.43×10^{-11}	4.45×10^{-11}	2.42×10^{-9}	3.78×10^{-10}	3.65×10^{-11}	2.99×10^{-11}	2.35×10^{-10}	2.42×10^{-9}
F31	3.02×10^{-11}	2.52×10^{-1}	8.12×10^{-4}	4.08×10^{-11}	2.52×10^{-1}	6.52×10^{-1}	2.15×10^{-2}	8.50×10^{-2}	9.92×10^{-11}	3.71×10^{-1}	9.92×10^{-11}
F32	3.02×10^{-11}	4.44×10^{-7}	5.61×10^{-5}	3.02×10^{-11}	6.01×10^{-8}	8.48×10^{-9}	1.07×10^{-7}	1.25×10^{-4}	3.02×10^{-11}	8.65×10^{-1}	3.02×10^{-11}
F33	3.02×10^{-11}	2.77×10^{-5}	5.07×10^{-10}	1.55×10^{-9}	1.36×10^{-7}	1.20×10^{-8}	6.01×10^{-8}	9.47×10^{-3}	3.02×10^{-11}	2.02×10^{-8}	3.87×10^{-1}
F34	3.02×10^{-11}	2.60×10^{-5}	2.83×10^{-8}	7.29×10^{-3}	3.03×10^{-3}	9.83×10^{-8}	7.17×10^{-1}	1.30×10^{-1}	3.02×10^{-11}	4.38×10^{-1}	8.53×10^{-1}
F35	3.02×10^{-11}	7.04×10^{-7}	8.99×10^{-11}	3.02×10^{-11}	1.61×10^{-10}	4.98×10^{-11}	1.89×10^{-4}	4.44×10^{-7}	3.02×10^{-11}	6.10×10^{-3}	6.07×10^{-11}
F36	3.34×10^{-11}	4.69×10^{-8}	1.09×10^{-5}	3.16×10^{-10}	9.26×10^{-9}	2.78×10^{-7}	9.71×10^{-1}	5.19×10^{-2}	3.02×10^{-11}	1.32×10^{-4}	2.24×10^{-2}
Hybrid											
F37	3.02×10^{-11}	4.50×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.34×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.37×10^{-5}
F38	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	7.20×10^{-5}
F39	3.02×10^{-11}	1.07×10^{-9}	3.02×10^{-11}	3.02×10^{-11}	4.50×10^{-11}	3.02×10^{-11}	5.00×10^{-9}	3.02×10^{-11}	3.02×10^{-11}	8.99×10^{-11}	2.92×10^{-2}
F40	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	4.68×10^{-2}
F41	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	2.32×10^{-2}
F42	3.02×10^{-11}	5.49×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	9.92×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	2.15×10^{-10}	3.51×10^{-2}

Table 9. Cont.

Fun	SSA	GBO	RUN	WOA	GTO	AVOA	EO	GWO	RSA	SMA	DE
Composition											
F43	1.20×10^{-11}	1.45×10^{-7}	1.45×10^{-7}	1.51×10^{-8}	1.45×10^{-7}	1.45×10^{-7}	2.43×10^{-11}	1.20×10^{-11}	1.45×10^{-7}	1.45×10^{-7}	6.39×10^{-4}
F44	1.86×10^{-9}	2.09×10^{-10}	3.34×10^{-11}	3.02×10^{-11}	1.79×10^{-11}	3.16×10^{-12}	2.38×10^{-7}	3.16×10^{-10}	3.16×10^{-12}	1.78×10^{-10}	2.23×10^{-9}
F45	7.77×10^{-9}	5.93×10^{-9}	6.93×10^{-12}	1.44×10^{-10}	4.57×10^{-12}	1.60×10^{-11}	1.34×10^{-10}	1.20×10^{-8}	1.02×10^{-11}	1.41×10^{-11}	2.53×10^{-8}
F46	5.97×10^{-5}	5.60×10^{-7}	3.67×10^{-3}	5.07×10^{-10}	1.29×10^{-6}	5.53×10^{-8}	1.49×10^{-6}	5.01×10^{-1}	3.02×10^{-11}	8.29×10^{-6}	2.13×10^{-5}
F47	2.25×10^{-4}	2.68×10^{-5}	2.76×10^{-7}	2.15×10^{-10}	1.50×10^{-5}	1.91×10^{-7}	4.80×10^{-7}	2.60×10^{-8}	2.32×10^{-7}	2.77×10^{-6}	1.35×10^{-4}
F48	3.56×10^{-4}	4.57×10^{-12}	4.57×10^{-12}	7.76×10^{-9}	4.57×10^{-12}	4.57×10^{-12}	3.82×10^{-9}	7.69×10^{-8}	4.57×10^{-12}	4.57×10^{-12}	6.51×10^{-9}
F49	3.02×10^{-11}	3.02×10^{-11}	1.17×10^{-9}	3.02×10^{-11}	6.52×10^{-9}	1.58×10^{-7}	3.02×10^{-11}	3.02×10^{-11}	5.89×10^{-2}	2.83×10^{-6}	3.02×10^{-11}
F50	3.02×10^{-11}	2.20×10^{-7}	1.47×10^{-7}	3.02×10^{-11}	3.82×10^{-10}	5.97×10^{-9}	5.60×10^{-7}	2.61×10^{-10}	1.21×10^{-12}	1.56×10^{-2}	8.50×10^{-2}

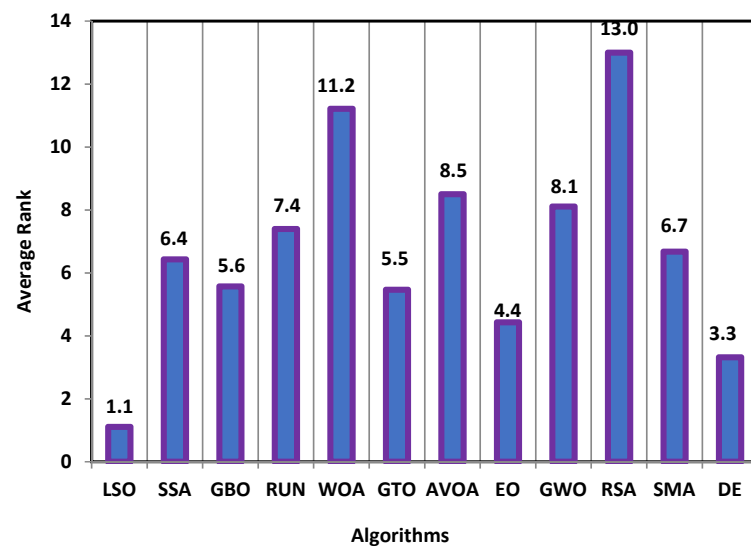


Figure 11. Average rank on all CEC2017 test functions.

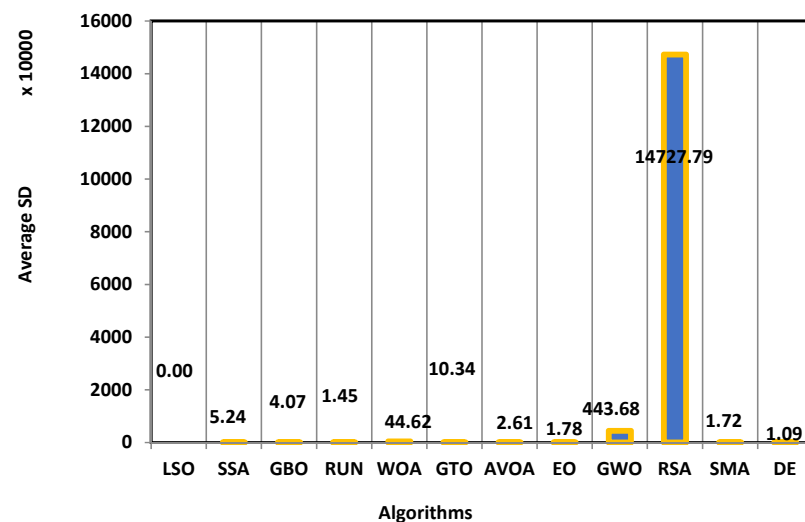


Figure 12. Average SD of each optimizer on all CEC2017.

The Wilcoxon rank-sum test is used to determine the difference between the outcomes of LSO and these of each rival optimizer on CEC2017 test functions. Wilcoxon rank-sum test demonstrates a significant difference between the outcomes of LSO with the rival algorithms, as the p -values in Table 11 support the alternative hypothesis. Ultimately, LSO is a strong optimizer, as demonstrated by its ability to defeat GBO, RUN, GTO, AVOA, SMA, RSA, and EO, which are the most recently published optimizers, as well as four highly-cited metaheuristic algorithms such as WOA, GWO, SSA, and DE.

Table 10. Avr and SD of CEC2017 for 25 independent runs (The bolded value is the best overall).

F	Index	LSO	SSA	GBO	RUN	WOA	GTO	AVOA	EO	GWO	RSA	SMA	DE
Unimodal													
F51	Avr	1.00×10^2	3.02×10^3	1.75×10^3	3.41×10^3	4.66×10^6	3.13×10^3	4.12×10^3	1.79×10^3	4.87×10^7	1.14×10^{10}	7.46×10^3	4.60×10^2
	SD	3.56×10^{-14}	3.59×10^3	1.71×10^3	1.77×10^3	6.92×10^6	3.24×10^3	4.04×10^3	1.93×10^3	1.26×10^8	3.64×10^9	4.48×10^3	1.19×10^3
	Rank	1.00	5.00	3.00	7.00	10.00	6.00	8.00	4.00	11.00	13.00	9.00	2.00
F52	Avr	300.00	300.00	300.00	300.00	1617.08	300.00	300.00	300.00	2542.88	10,392.16	300.00	300.00
	SD	0.00×10^0	7.30×10^{-10}	4.88×10^{-13}	1.17×10^{-3}	1.73×10^3	4.70×10^{-10}	2.93×10^{-9}	4.92×10^{-10}	2.02×10^3	3.07×10^3	1.62×10^{-3}	1.06×10^{-14}
	Rank	1.00	6.00	3.00	9.00	11.00	4.00	7.00	5.00	12.00	13.00	8.00	2.00
Multimodal													
F53	Avr	400.00	405.28	401.36	403.35	428.61	402.75	408.82	403.49	418.77	1226.25	408.54	406.25
	SD	6.83×10^{-7}	10.06	0.49	2.18	36.66	1.38	17.51	1.28	20.57	579.17	13.53	1.24
	Rank	1.00	6.00	2.00	4.00	11.00	3.00	9.00	5.00	10.00	13.00	8.00	7.00
F54	Avr	506.00	519.80	526.09	533.13	555.93	526.93	537.46	516.16	519.42	584.87	514.30	514.54
	SD	2.22	9.38	12.38	8.51	26.06	12.75	14.36	5.66	8.78	15.78	6.18	3.31
	Rank	1.00	6.00	7.00	9.00	12.00	8.00	10.00	4.00	5.00	13.00	2.00	3.00
F55	Avr	600.00	609.86	601.84	618.91	638.19	609.14	610.07	600.06	601.57	645.54	600.15	600.00
	SD	0.00	7.52×10^0	3.06×10^0	8.59×10^0	1.37×10^1	7.16×10^0	6.92×10^0	3.17×10^{-1}	2.14×10^0	6.58×10^0	4.35×10^{-1}	2.08×10^{-6}
	Rank	1.00	8.00	6.00	10.00	12.00	7.00	9.00	3.00	5.00	13.00	4.00	2.00
F56	Avr	717.07	732.33	737.11	762.79	775.78	756.31	765.61	723.21	732.39	806.57	726.05	726.99
	SD	2.58	9.87	11.64	12.14	23.43	17.12	20.50	5.84	8.53	12.89	7.54	3.86
	Rank	1.00	5.00	7.00	9.00	12.00	8.00	10.00	2.00	6.00	13.00	3.00	4.00
F57	Avr	806.62	821.03	822.97	827.43	840.82	824.91	828.43	813.33	815.53	854.98	816.32	814.53
	SD	3.42	10.61	8.06	6.00	16.83	9.14	10.25	4.85	5.62	7.67	7.52	3.69
	Rank	1.00	6.00	7.00	9.00	12.00	8.00	10.00	2.00	4.00	13.00	5.00	3.00
F58	Avr	900.00	903.92	924.99	1056.95	1498.95	993.42	1090.74	900.20	927.36	1542.77	900.00	900.00
	SD	0.00	7.74×10^0	4.43×10^1	1.01×10^2	4.48×10^2	8.62×10^1	1.80×10^2	5.29×10^{-1}	5.44×10^1	1.67×10^2	1.64×10^{-2}	0.00
	Rank	1.00	5.00	6.00	10.00	12.00	8.00	11.00	4.00	7.00	13.00	3.00	1.00
F59	Avr	1350.60	1772.60	1835.55	1582.06	2158.15	2000.11	1852.32	1541.39	1639.92	2623.40	1601.13	1449.25
	SD	143.76	259.50	342.45	241.56	392.70	311.11	312.11	222.43	314.70	171.28	224.02	262.92
	Rank	1.00	7.00	8.00	4.00	11.00	10.00	9.00	3.00	6.00	13.00	5.00	2.00
Hybrid													
F60	Avr	1101.17	2158.63	1120.62	1126.24	1378.55	1126.78	1138.46	1108.96	1141.03	19,833.64	1116.24	1102.18
	SD	1.16	464.08	27.18	7.71	443.63	17.50	39.91	8.28	36.04	48,865.27	7.73	2.31
	Rank	1.00	12.00	5.00	6.00	10.00	7.00	8.00	3.00	9.00	13.00	4.00	2.00

Table 10. Cont.

F	Index	LSO	SSA	GBO	RUN	WOA	GTO	AVOA	EO	GWO	RSA	SMA	DE
F61	Avr	1233.50	1.20×10^6	1.06×10^4	2.18×10^5	4.62×10^6	1.40×10^4	3.87×10^5	1.01×10^4	6.84×10^5	2.75×10^8	4.51×10^4	3.83×10^3
	SD	52.56	9.97×10^5	1.25×10^4	1.87×10^5	5.01×10^6	1.22×10^4	4.73×10^5	6.61×10^3	8.47×10^5	1.59×10^8	2.89×10^4	6.45×10^3
	Rank	1.00	10.00	4.00	7.00	11.00	5.00	8.00	3.00	9.00	13.00	6.00	2.00
F62	Avr	1303.75	16,604.03	1815.02	10,568.95	16,564.10	1466.29	10,902.04	7359.12	11,669.47	49,487,747.88	15,922.68	1309.37
	SD	2.55	11,742.29	304.94	5329.62	13,327.54	149.89	8235.90	6429.88	6780.52	77,696,384.86	13,216.90	7.75
	Rank	1.00	11.00	4.00	6.00	10.00	3.00	7.00	5.00	8.00	13.00	9.00	2.00
F63	Avr	1400.60	1498.92	1480.73	2086.66	1803.91	1457.81	1790.54	1455.40	2955.58	10,856.69	1581.31	1402.50
	SD	0.62	40.21	38.32	589.06	728.21	24.43	554.19	27.24	1781.83	18,963.95	642.53	5.99
	Rank	1.00	6.00	5.00	11.00	10.00	4.00	9.00	3.00	12.00	13.00	7.00	2.00
F64	Avr	1500.29	2251.04	1593.26	2267.56	6197.31	1559.16	3328.43	1600.40	4020.66	11,243.41	2344.58	1500.66
	SD	0.34	883.94	74.16	783.57	4167.90	52.61	1537.34	76.54	3824.80	5685.89	1528.88	0.61
	Rank	1.00	7.00	4.00	8.00	12.00	3.00	10.00	5.00	11.00	13.00	9.00	2.00
F65	Avr	1600.46	1709.98	1740.23	1783.44	1899.10	1701.90	1797.66	1664.51	1725.36	2127.09	1691.78	1611.95
	SD	0.26	117.63	107.09	108.96	116.03	85.88	123.84	66.64	107.02	117.70	78.57	25.13
	Rank	1.00	6.00	9.00	10.00	12.00	5.00	11.00	3.00	8.00	13.00	4.00	2.00
F66	Avr	1702.47	1770.91	1761.37	1758.72	1793.83	1750.44	1780.81	1742.65	1749.74	1849.72	1756.58	1706.13
	SD	4.52	41.62	43.18	21.79	44.20	21.82	46.96	17.18	20.45	50.83	38.02	8.25
	Rank	1.00	9.00	8.00	7.00	12.00	5.00	11.00	3.00	4.00	13.00	6.00	2.00
F67	Avr	1800.18	1.63×10^4	4.67×10^3	1.68×10^4	1.76×10^4	3.13×10^3	1.40×10^4	1.42×10^4	2.64×10^4	1.55×10^8	2.80×10^4	1.81×10^3
	SD	0.27	1.01×10^4	4.95×10^3	9.94×10^3	1.31×10^4	2.79×10^3	9.74×10^3	1.08×10^4	1.63×10^4	3.78×10^8	1.42×10^4	8.76×10^0
	Rank	1.00	7.00	4.00	8.00	9.00	3.00	5.00	6.00	10.00	13.00	11.00	2.00
F68	Avr	1900.11	2551.75	1987.39	8849.78	52,403.86	1955.14	7572.26	1954.94	15,942.55	2,103,955.73	6030.92	1900.30
	SD	0.27	886.53	77.78	6841.59	85,717.24	49.95	6392.57	34.09	47,387.19	3,083,821.04	6113.48	0.55
	Rank	1.00	6.00	5.00	10.00	12.00	4.00	9.00	3.00	11.00	13.00	7.00	2.00
F69	Avr	2000.23	2088.91	2093.27	2125.33	2205.96	2082.82	2098.95	2055.37	2081.95	2300.60	2025.85	2002.40
	SD	0.38	57.36	63.21	54.09	67.36	50.96	75.83	61.47	60.40	73.00	9.52	6.32
	Rank	1.00	7.00	8.00	11.00	12.00	6.00	10.00	4.00	5.00	13.00	3.00	2.00
Composition													
F70	Avr	2215.36	2276.54	2274.36	2233.44	2306.01	2222.91	2252.57	2312.71	2317.95	2337.09	2302.12	2308.46
	SD	34.45	58.28	60.70	54.68	64.60	48.29	67.13	22.59	5.76	59.07	45.92	36.96
	Rank	1.00	7.00	6.00	3.00	9.00	2.00	5.00	11.00	12.00	13.00	8.00	10.00
F71	Avr	2291.09	2350.02	2300.86	2305.55	2579.66	2302.98	2337.24	2316.37	2348.58	3123.29	2387.92	2300.47
	SD	24.29	177.41	13.46	11.34	536.23	10.16	192.17	85.77	152.10	345.67	286.27	0.80
	Rank	1.00	9.00	3.00	5.00	12.00	4.00	7.00	6.00	8.00	13.00	11.00	2.00

Table 10. Cont.

F	Index	LSO	SSA	GBO	RUN	WOA	GTO	AVOA	EO	GWO	RSA	SMA	DE
F72	Avr	2597.61	2620.27	2624.73	2620.77	2646.73	2631.16	2640.09	2615.40	2618.59	2701.83	2620.81	2612.69
	SD	56.22	8.88	12.93	6.95	21.63	17.77	17.08	5.37	7.67	22.14	8.01	4.21
	Rank	1.00	5.00	8.00	6.00	11.00	9.00	10.00	3.00	4.00	13.00	7.00	2.00
F73	Avr	2650.08	2749.50	2698.60	2740.68	2765.21	2723.20	2735.38	2744.79	2748.32	2872.14	2755.56	2751.45
	SD	112.94	9.32	112.23	46.31	62.33	89.82	94.83	7.51	10.93	44.62	10.18	5.29
	Rank	1.00	8.00	2.00	5.00	11.00	3.00	4.00	6.00	7.00	13.00	10.00	9.00
F74	Avr	2902.82	2921.26	2934.68	2935.62	2938.57	2919.17	2927.29	2930.31	2938.18	3349.41	2945.50	2929.57
	SD	13.79	24.61	29.09	20.38	63.39	66.92	24.71	22.34	28.39	117.10	33.08	22.53
	Rank	1.00	3.00	7.00	8.00	10.00	2.00	4.00	6.00	9.00	13.00	11.00	5.00
F75	Avr	2845.98	2975.91	2980.81	2993.65	3345.53	2989.79	3250.79	2973.06	3263.92	4146.59	3176.37	3001.41
	SD	103.36	249.95	114.69	213.33	446.62	129.74	529.36	253.79	442.08	290.19	427.89	231.12
	Rank	1.00	3.00	4.00	6.00	12.00	5.00	10.00	2.00	11.00	13.00	9.00	7.00
F76	Avr	3090.28	3092.86	3097.64	3094.37	3141.37	3099.69	3101.52	3095.37	3099.26	3189.41	3091.13	3092.04
	SD	1.78	3.08	4.37	2.27	37.19	18.88	8.87	8.41	12.72	84.08	1.51	3.12
	Rank	1.00	4.00	7.00	5.00	12.00	9.00	10.00	6.00	8.00	13.00	2.00	3.00
F77	Avr	3111.62	3312.50	3315.61	3340.49	3433.65	3317.49	3324.96	3342.76	3335.87	3780.91	3367.88	3334.58
	SD	150.21	185.24	138.98	108.50	181.80	188.84	128.45	122.45	112.63	145.33	164.48	108.44
	Rank	1.00	3.00	4.00	9.00	12.00	5.00	6.00	10.00	8.00	13.00	11.00	7.00
F78	Avr	3157.42	3185.14	3246.87	3206.20	3327.95	3220.80	3278.28	3192.92	3213.86	3463.17	3198.95	3170.47
	SD	9.57	43.95	73.76	36.66	74.48	48.13	79.14	48.33	48.58	140.12	67.35	16.12
	Rank	1.00	3.00	10.00	6.00	12.00	8.00	11.00	4.00	7.00	13.00	5.00	2.00
F79	Avr	3.49×10^3	3.12×10^5	8.06×10^5	1.67×10^5	9.11×10^5	1.90×10^6	1.64×10^5	3.38×10^5	9.62×10^5	9.67×10^6	2.33×10^5	1.48×10^5
	SD	2.20×10^2	4.94×10^5	1.16×10^6	2.06×10^5	8.87×10^5	2.98×10^6	2.50×10^5	4.89×10^5	1.74×10^6	1.32×10^7	4.27×10^5	3.07×10^5
	Rank	1.00	6.00	9.00	4.00	10.00	12.00	3.00	7.00	11.00	13.00	5.00	2.00

Table 11. *p*-values of LSO with each rival optimizer on CEC-2017 test suite (F51–F79).

Fun	SSA	GBO	RUN	WOA	GTO	AVOA	EO	GWO	RSA	SMA	DE
Unimodal											
F51	2.40×10^{-11}	2.40×10^{-11}	2.40×10^{-11}	2.40×10^{-11}	2.40×10^{-11}	2.40×10^{-11}	2.40×10^{-11}	2.40×10^{-11}	2.40×10^{-11}	2.40×10^{-11}	2.92×10^{-1}
F52	1.21×10^{-12}	1.53×10^{-11}	1.21×10^{-12}	1.21×10^{-12}	1.20×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	3.34×10^{-1}
Multimodal											
F53	3.33×10^{-11}	3.33×10^{-11}	4.96×10^{-11}	3.01×10^{-11}	3.68×10^{-11}	4.49×10^{-11}	3.01×10^{-11}	3.01×10^{-11}	3.01×10^{-11}	3.01×10^{-11}	3.01×10^{-11}
F54	2.60×10^{-8}	3.47×10^{-10}	3.69×10^{-11}	3.34×10^{-11}	4.20×10^{-10}	4.50×10^{-11}	3.65×10^{-8}	7.12×10^{-9}	3.02×10^{-11}	2.00×10^{-6}	8.48×10^{-9}
F55	2.36×10^{-12}	2.36×10^{-12}	2.36×10^{-12}	2.36×10^{-12}	2.36×10^{-12}	2.36×10^{-12}	4.48×10^{-12}	2.36×10^{-12}	2.36×10^{-12}	2.36×10^{-12}	4.02×10^{-1}
F56	2.61×10^{-10}	9.92×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	6.28×10^{-6}	1.69×10^{-9}	3.02×10^{-11}	5.19×10^{-7}	2.61×10^{-10}
F57	1.25×10^{-7}	1.96×10^{-10}	3.02×10^{-11}	3.02×10^{-11}	4.62×10^{-10}	4.97×10^{-11}	5.17×10^{-7}	1.20×10^{-8}	3.02×10^{-11}	2.19×10^{-8}	4.57×10^{-9}
F58	5.26×10^{-12}	8.63×10^{-12}	1.72×10^{-12}	1.72×10^{-12}	1.72×10^{-12}	1.72×10^{-12}	4.09×10^{-11}	2.15×10^{-12}	1.72×10^{-12}	4.10×10^{-11}	3.34×10^{-1}
F59	2.19×10^{-8}	2.19×10^{-8}	8.66×10^{-5}	1.96×10^{-10}	2.15×10^{-10}	6.52×10^{-9}	9.03×10^{-4}	1.78×10^{-4}	3.02×10^{-11}	9.51×10^{-6}	2.40×10^{-1}
Hybrid											
F60	3.02×10^{-11}	9.92×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.69×10^{-11}	3.02×10^{-11}	1.31×10^{-8}	3.02×10^{-11}	3.02×10^{-11}	4.50×10^{-11}	1.08×10^{-2}
F61	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}
F62	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	1.60×10^{-7}
F63	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	1.55×10^{-1}
F64	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	6.35×10^{-2}
F65	3.02×10^{-11}	6.70×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	4.08×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	7.09×10^{-8}
F66	3.02×10^{-11}	4.08×10^{-11}	3.69×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	5.49×10^{-11}	3.69×10^{-11}	3.02×10^{-11}	4.50×10^{-11}	2.43×10^{-1}
F67	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	2.61×10^{-2}
F68	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	1.76×10^{-2}
F69	1.41×10^{-11}	1.41×10^{-11}	1.41×10^{-11}	1.41×10^{-11}	1.41×10^{-11}	1.41×10^{-11}	2.61×10^{-11}	1.41×10^{-11}	1.41×10^{-11}	1.41×10^{-11}	5.22×10^{-2}
Composition											
F70	1.32×10^{-4}	1.32×10^{-4}	4.46×10^{-1}	4.44×10^{-7}	9.94×10^{-1}	4.86×10^{-3}	1.70×10^{-8}	1.96×10^{-10}	4.44×10^{-7}	9.26×10^{-9}	3.08×10^{-8}
F71	3.34×10^{-11}	1.55×10^{-9}	3.82×10^{-10}	3.82×10^{-10}	1.17×10^{-9}	4.31×10^{-8}	6.77×10^{-5}	4.98×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	8.07×10^{-1}

Table 11. Cont.

Fun	SSA	GBO	RUN	WOA	GTO	AVOA	EO	GWO	RSA	SMA	DE
F72	2.67×10^{-9}	5.49×10^{-11}	1.21×10^{-10}	3.02×10^{-11}	1.07×10^{-9}	3.34×10^{-11}	8.48×10^{-9}	3.96×10^{-8}	3.02×10^{-11}	4.50×10^{-11}	4.74×10^{-6}
F73	2.60×10^{-8}	1.76×10^{-3}	2.38×10^{-7}	2.23×10^{-9}	3.32×10^{-6}	8.20×10^{-7}	2.32×10^{-6}	7.04×10^{-7}	3.02×10^{-11}	1.61×10^{-10}	2.87×10^{-10}
F74	4.57×10^{-5}	2.53×10^{-8}	8.61×10^{-10}	3.86×10^{-8}	4.66×10^{-6}	1.44×10^{-7}	4.44×10^{-9}	5.30×10^{-9}	2.91×10^{-11}	7.51×10^{-8}	5.00×10^{-6}
F75	1.77×10^{-9}	8.48×10^{-8}	2.88×10^{-2}	1.80×10^{-10}	6.34×10^{-7}	4.13×10^{-6}	2.12×10^{-6}	3.55×10^{-10}	1.44×10^{-11}	3.64×10^{-11}	7.62×10^{-7}
F76	6.32×10^{-5}	6.63×10^{-10}	2.80×10^{-8}	2.98×10^{-11}	3.32×10^{-8}	1.39×10^{-9}	3.80×10^{-6}	1.99×10^{-8}	2.98×10^{-11}	3.17×10^{-3}	4.20×10^{-4}
F77	3.78×10^{-8}	7.35×10^{-6}	2.09×10^{-9}	1.73×10^{-9}	1.24×10^{-6}	2.28×10^{-8}	2.49×10^{-7}	2.09×10^{-8}	2.50×10^{-11}	3.60×10^{-9}	2.40×10^{-8}
F78	5.87×10^{-4}	7.38×10^{-10}	9.26×10^{-9}	3.02×10^{-11}	4.18×10^{-9}	6.07×10^{-11}	1.86×10^{-6}	5.46×10^{-9}	3.02×10^{-11}	3.51×10^{-2}	9.03×10^{-4}
F79	3.02×10^{-11}	5.07×10^{-10}	3.02×10^{-11}	3.02×10^{-11}	5.46×10^{-11}	3.02×10^{-11}	3.69×10^{-11}	3.02×10^{-11}	3.01×10^{-11}	3.02×10^{-11}	2.87×10^{-10}

4.6. LSO for Challengeable CEC2020

During this section, additional testing is carried out on the CEC-2020 test suite to determine whether the proposed has stable performance for more challenging test functions. An algorithm's ability to explore, exploit, and stay away from local minima can be evaluated using this suite, which consists of ten test functions and is divided into four categories: unimodal, multimodal, hybrid, and compositional. The characteristics of this suite are shown in Appendix A Tables A1–A7. LSO has superior performance for all test functions found in the CEC-2020 test suite except for F83, as evidenced by the *Avr*, rank, and SD values presented in Table 12 and obtained after analyzing the results of 25 independent runs. Figures 13 and 14 show the average of the rank values and SD on all test functions of CEC2020. According to these figures, LSO is the best because it is ranked first with a value of 1.7 and has the lowest standard deviation of 38, whereas RSA is the worst because it is ranked last with a value of 12. Finally, the Wilcoxon rank-sum test is used to determine the difference between the results of LSO and those of each rival optimizer on this suite. The Wilcoxon rank-sum test results demonstrate a statistically significant difference between the outcomes of LSO and the rival algorithms for most test functions, as evidenced by the *p*-values in Table 13, which support the alternative hypothesis. As an added bonus, in this section, we present additional experimental evidence that LSO belongs to the strong optimizers.

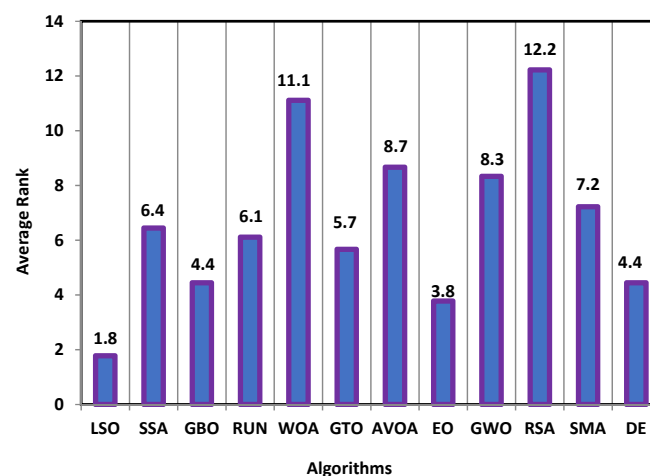


Figure 13. Average rank of each optimizer on all CEC2020.

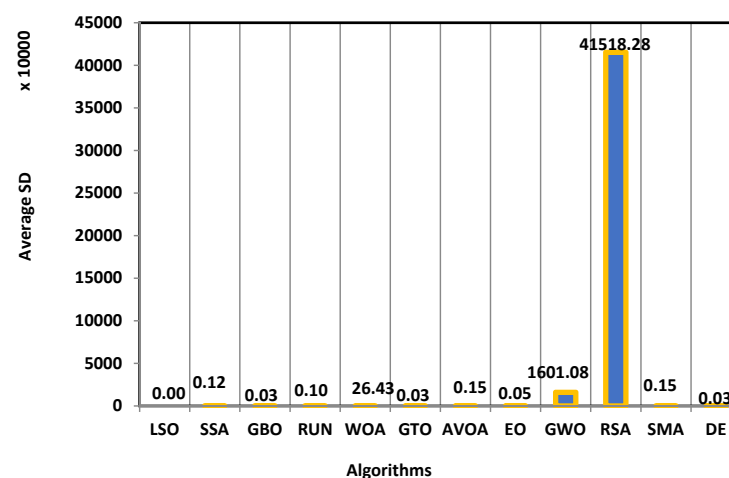


Figure 14. Average SD of each optimizer on all CEC2020.

Table 12. Avr and SD of CEC2020 for 25 independent runs (The bolded value is the best overall).

F	Index	LSO	SSA	GBO	RUN	WOA	GTO	AVOA	EO	GWO	RSA	SMA	DE
Unimodal													
F80	Avr	1.00×10^2	3.04×10^3	1.82×10^3	3.94×10^3	1.69×10^6	2.85×10^3	3.76×10^3	3.89×10^3	7.74×10^7	$1.27 \times 10^{+10}$	7257.82	704.92
	SD	2.64×10^{-13}	3.47×10^3	2.05×10^3	2.04×10^3	2.04×10^6	2.19×10^3	2.52×10^3	3.64×10^3	1.60×10^8	4.15×10^9	4390.97	2059.48
	Rank	1.00	5.00	3.00	8.00	10.00	4.00	6.00	7.00	11.00	13.00	9.00	2.00
Multimodal													
F81	Avr	1347.02	1766.98	1839.45	1636.20	2059.68	1954.49	1902.30	1476.48	1682.49	2648.60	1665.28	1604.18
	SD	1.52×10^2	2.87×10^2	3.42×10^2	1.98×10^2	3.46×10^2	2.39×10^2	3.00×10^2	1.92×10^2	2.48×10^2	2.37×10^2	158.68	251.92
	Rank	1.00	7.00	8.00	4.00	11.00	10.00	9.00	2.00	6.00	13.00	5.00	3.00
Hybrid													
F82	Avr	718.49	732.59	744.51	761.38	784.64	758.14	769.60	725.86	732.62	806.17	727.48	726.59
	SD	3.37	9.83	15.30	17.00	21.87	18.01	19.84	7.84	10.45	10.00	9.09	3.31
	Rank	1.00	5.00	7.00	9.00	12.00	8.00	10.00	2.00	6.00	13.00	4.00	3.00
F83	Avr	1900.02	1901.35	1900.00	1900.00	1900.11	1900.00	1900.00	1900.00	1900.11	1900.00	1900.00	1901.16
	SD	0.09	0.56	0.00	0.00	0.46	0.00	0.00	0.00	0.26	0.00	0.00	0.17
	Rank	8.00	13.00	1.00	2.00	9.00	3.00	4.00	5.00	10.00	6.00	7.00	12.00
F84	Avr	1712.49	6902.12	2245.58	6702.53	294,472.06	2278.87	7280.40	3432.45	39,529.67	414,990.21	9364.45	1834.94
	SD	7.64×10^0	4.34×10^3	2.81×10^2	3.78×10^3	5.64×10^5	3.21×10^2	5.18×10^3	9.92×10^2	1.03×10^5	1.37×10^5	5902.78	239.86
	Rank	1.00	7.00	3.00	6.00	12.00	4.00	8.00	5.00	11.00	13.00	9.00	2.00
F85	Avr	1604.58	1713.70	1746.69	1756.90	1824.55	1776.96	1788.64	1671.19	1747.52	2137.60	1722.52	1618.83
	SD	21.66	87.85	97.05	77.33	115.43	101.48	106.80	94.61	87.98	213.01	52.60	47.72
	Rank	1.00	4.00	6.00	8.00	12.00	9.00	11.00	3.00	7.00	13.00	5.00	2.00
F86	Avr	2100.92	5038.57	2561.38	5704.18	45,897.21	2546.39	8936.41	2417.44	9055.12	1,045,287.18	5349.38	2115.06
	SD	3.01	3364.65	263.40	3932.69	38,048.22	354.82	6470.47	208.65	4287.62	1,690,081.52	4114.90	51.19
	Rank	1.00	6.00	5.00	8.00	12.00	4.00	9.00	3.00	10.00	13.00	7.00	2.00

Table 12. Cont.

F	Index	LSO	SSA	GBO	RUN	WOA	GTO	AVOA	EO	GWO	RSA	SMA	DE
Composition													
F87	Avr	2291.63	2298.76	2299.33	2305.76	2401.00	2305.77	2336.93	2294.71	2308.68	3188.89	2413.02	2300.39
	SD	2.33×10^1	1.93×10^1	1.74×10^1	1.09×10^1	2.60×10^2	4.59×10^0	1.82×10^2	2.32×10^1	2.43×10^1	2.97×10^2	308.22	0.56
	Rank	1.00	3.00	4.00	6.00	11.00	7.00	9.00	2.00	8.00	13.00	12.00	5.00
F88	Avr	2654.71	2748.83	2717.24	2740.74	2774.76	2707.03	2777.68	2742.18	2745.64	2860.81	2746.39	2749.71
	SD	113.82	8.28	99.81	46.31	57.90	105.77	22.49	6.95	48.68	34.68	47.52	4.42
	Rank	1.00	8.00	3.00	4.00	11.00	2.00	12.00	5.00	6.00	13.00	7.00	9.00
F89	Avr	2894.33	2922.20	2937.04	2929.85	2952.00	2926.60	2931.74	2932.25	2936.16	3372.91	2927.64	2936.00
	SD	57.51	24.00	22.48	23.57	23.42	24.33	23.04	22.98	18.18	150.11	31.06	19.13
	Rank	1.00	2.00	10.00	5.00	11.00	3.00	6.00	7.00	9.00	13.00	4.00	8.00

Table 13. *p*-values of LSO with each rival optimizer on CEC-2020 test suite (F80–F89).

F	SSA	GBO	RUN	WOA	GTO	AVOA	EO	GWO	RSA	SMA	DE
Unimodal											
F80	2.61×10^{-11}	2.61×10^{-11}	2.61×10^{-11}	2.61×10^{-11}	2.61×10^{-11}	2.61×10^{-11}	2.61×10^{-11}	2.61×10^{-11}	2.61×10^{-11}	2.61×10^{-11}	3.49×10^{-1}
Multimodal											
F81	9.06×10^{-8}	9.83×10^{-8}	7.60×10^{-7}	2.61×10^{-10}	2.87×10^{-10}	2.67×10^{-9}	1.22×10^{-2}	5.60×10^{-7}	3.02×10^{-11}	3.65×10^{-8}	7.66×10^{-5}
Hybrid											
F82	2.67×10^{-9}	8.15×10^{-11}	3.34×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	8.15×10^{-5}	6.53×10^{-8}	3.02×10^{-11}	1.34×10^{-5}	1.41×10^{-9}
F83	1.72×10^{-12}	3.34×10^{-1}	3.34×10^{-1}	5.44×10^{-1}	3.34×10^{-1}	3.34×10^{-1}	3.34×10^{-1}	2.12×10^{-4}	3.34×10^{-1}	3.34×10^{-1}	1.72×10^{-12}
F84	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	5.46×10^{-6}
F85	1.09×10^{-10}	6.70×10^{-11}	4.50×10^{-11}	4.08×10^{-11}	4.98×10^{-11}	5.49×10^{-11}	6.72×10^{-10}	6.07×10^{-11}	3.02×10^{-11}	4.98×10^{-11}	1.70×10^{-2}
F86	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.34×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	8.31×10^{-3}
Composition											
F87	6.52×10^{-9}	3.35×10^{-8}	3.82×10^{-10}	3.02×10^{-11}	3.02×10^{-11}	4.18×10^{-9}	2.89×10^{-3}	5.07×10^{-10}	3.02×10^{-11}	6.52×10^{-9}	4.19×10^{-1}
F88	8.48×10^{-9}	2.28×10^{-5}	5.53×10^{-8}	2.61×10^{-10}	1.11×10^{-4}	3.02×10^{-11}	4.98×10^{-4}	2.60×10^{-5}	3.02×10^{-11}	2.92×10^{-9}	2.37×10^{-10}
F89	1.39×10^{-5}	2.49×10^{-8}	3.64×10^{-9}	4.37×10^{-10}	2.03×10^{-7}	8.84×10^{-9}	1.66×10^{-7}	1.77×10^{-8}	2.84×10^{-11}	2.24×10^{-6}	4.79×10^{-8}

4.7. LSO for Challengeable CEC2022

The proposed and other methods are tested again on the CEC2022 test suite. This test suite contains 12 test functions divided into unimodal, multimodal, hybrid, and compositional. The properties of this test suite are also listed in Appendix A (Tables A1–A7), and their dimensions in the conducted experiments are 10. Table 14 shows the Avr, rank, and SD for 25 independent runs, demonstrating LSO's superior performance for 9 out of 12 test functions of the CEC-2022 test suite. The average of the rank values and standard deviation on all test functions of CEC2022 are depicted in Figures 15 and 16. These figures show that LSO is the best because it is ranked first with a value of 1.6 and has the lowest standard deviation of 12, whereas RSA is the worst because it is ranked last with a value of 12 and has the highest standard deviation. In the end, the Wilcoxon rank-sum test is used to determine whether there is a significant difference between the results of LSO and those of each rival optimizer on this suite of problems. For most test functions, the Wilcoxon rank-sum test results demonstrate a statistically significant difference between the outcomes of LSO and the rival algorithms, as demonstrated by the *p*-values in Table 15. This section further affirms that LSO belongs to the category of highly-performed optimizers.

Table 14. Avr and SD of CEC2022 for 25 independent runs (The bolded value is the best overall).

F	Index	LSO	SSA	GBO	RUN	WOA	GTO	AVOA	EO	GWO	RSA	SMA	DE
Unimodal													
F90	Avr	300.00	3.00×10^2	3.00×10^2	3.00×10^2	1.55×10^4	3.00×10^2	3.00×10^2	3.00×10^2	2.44×10^3	10,731.57	300.00	300.00
	SD	0.00	5.61×10^{-10}	1.37×10^{-10}	2.20×10^{-1}	6.57×10^3	6.31×10^{-9}	2.00×10^{-8}	6.18×10^{-10}	2.20×10^3	3788.59	0.00	0.00
	Rank	1.00	5.00	3.00	9.00	13.00	6.00	7.00	4.00	11.00	12.00	1.00	1.00
Multimodal													
F91	Avr	400.96	410.29	405.42	405.90	428.03	410.25	413.99	409.75	434.34	1066.53	409.80	407.52
	SD	2.13×10^0	1.96×10^1	3.66×10^0	1.30×10^1	4.56×10^1	1.74×10^1	2.22×10^1	1.71×10^1	2.63×10^1	814.53	12.35	2.32
	Rank	1.00	8.00	2.00	3.00	10.00	7.00	9.00	5.00	11.00	13.00	6.00	4.00
F92	Avr	600.00	606.09	601.30	617.36	632.26	609.38	611.84	600.15	601.21	645.65	600.08	600.00
	SD	0.00	4.62	1.85	7.40	13.11	7.56	6.30	0.62	1.86	6.13	0.05	2.07×10^{-6}
	Rank	1.00	7.00	6.00	10.00	12.00	8.00	9.00	4.00	5.00	13.00	3.00	2.00
F93	Avr	810.72	821.06	821.13	823.32	836.29	824.64	831.03	812.11	814.64	848.55	825.60	824.02
	SD	2.96	10.98	9.21	6.31	16.19	6.72	8.71	5.90	8.58	6.58	8.37	3.22
	Rank	1.00	4.00	5.00	6.00	11.00	8.00	10.00	2.00	3.00	13.00	9.00	7.00
F94	Avr	900.02	908.19	921.08	1021.89	1447.74	1019.31	1210.33	900.38	917.15	1432.77	902.70	900.00
	SD	8.29×10^{-2}	2.65×10^1	2.66×10^1	7.76×10^1	4.10×10^2	1.10×10^2	1.75×10^2	4.17×10^{-1}	2.84×10^1	104.11	8.25	1.60×10^{-6}
	Rank	2.00	5.00	7.00	10.00	13.00	9.00	11.00	3.00	6.00	12.00	4.00	1.00
Hybrid													
F95	Avr	1800.36	3566.29	2258.98	3181.78	3989.73	1874.01	3629.15	3691.90	5032.02	2.66×10^8	5411.36	1826.60
	SD	0.34	1735.60	1245.49	1115.74	1954.69	50.56	1615.83	1945.92	2475.45	2.41×10^8	2210.90	117.17
	Rank	1.00	6.00	4.00	5.00	9.00	3.00	7.00	8.00	10.00	1.30×10^1	11.00	2.00
F96	Avr	2000.35	2033.78	2027.67	2042.25	2066.69	2032.20	2033.24	2032.42	2033.04	2139.27	2020.43	2002.21
	SD	0.54	13.08	9.40	11.42	28.19	16.41	11.51	38.20	15.80	41.11	3.76	6.18
	Rank	1.00	9.00	4.00	10.00	12.00	5.00	8.00	6.00	7.00	13.00	3.00	2.00

Table 14. Cont.

F	Index	LSO	SSA	GBO	RUN	WOA	GTO	AVOA	EO	GWO	RSA	SMA	DE
F97	Avr	2201.57	2223.04	2221.27	2224.21	2232.31	2221.58	2224.61	2222.82	2224.73	2263.95	2220.57	2204.54
	SD	3.83×10^0	7.12×10^0	1.01×10^0	1.34×10^0	5.42×10^0	2.81×10^0	2.93×10^0	2.33×10^1	3.53×10^0	29.43	3.76	8.49
	Rank	1.00	7.00	4.00	8.00	12.00	5.00	9.00	6.00	10.00	13.00	3.00	2.00
Composition													
F98	Avr	2529.28	2529.43	2534.18	2529.29	2564.67	2529.55	2534.18	2529.28	2580.54	2771.04	2529.28	2529.28
	SD	0.00	0.19	26.83	0.00	39.90	0.87	26.83	0.00	39.65	61.34	0.00	6.79×10^{-10}
	Rank	1.00	6.00	8.00	5.00	11.00	7.00	9.00	2.00	12.00	13.00	4.00	3.00
F99	Avr	2500.36	2508.72	2516.28	2553.99	2616.35	2532.77	2558.84	2553.16	2573.33	2690.10	2523.74	2527.05
	SD	0.06	32.01	40.44	57.97	214.36	54.42	63.67	60.40	63.44	195.95	53.37	49.31
	Rank	1.00	3.00	4.00	9.00	12.00	7.00	10.00	8.00	11.00	13.00	5.00	6.00
F100	Avr	2825.01	2.82×10^3	2.78×10^3	2.70×10^3	2.92×10^3	<u>2.68×10^3</u>	2.78×10^3	2.85×10^3	3.00×10^3	4092.28	2750.51	2895.74
	SD	129.15	2.24×10^2	1.76×10^2	1.38×10^2	5.55×10^1	<u>1.27×10^2</u>	1.48×10^2	1.15×10^2	1.74×10^2	331.37	175.84	27.73
	Rank	7.00	6.00	5.00	2.00	11.00	1.00	4.00	8.00	12.00	13.00	3.00	9.00
F101	Avr	2862.69	2863.95	2865.40	2864.29	2894.24	2864.79	2866.37	2863.94	2866.00	2971.50	2862.46	2863.56
	SD	1.72	1.49	1.80	1.78	35.17	1.76	3.73	1.78	5.67	114.51	1.67	1.45
	Rank	2.00	5.00	8.00	6.00	12.00	7.00	10.00	4.00	9.00	13.00	1.00	3.00

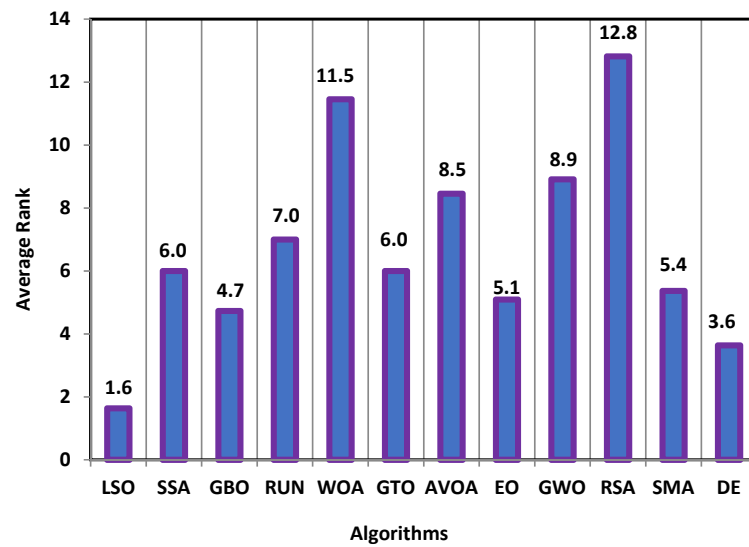


Figure 15. Average rank of each optimizer on all CEC2022.

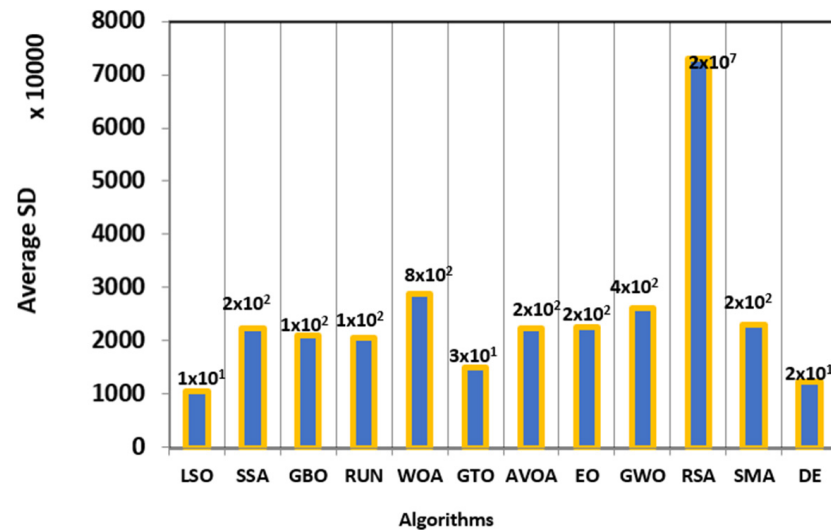


Figure 16. Average SD of each optimizer on all CEC2022.

Table 15. *p*-values of LSO with each rival optimizer on CEC-2022 test suite (F90–F101).

F	SSA	GBO	RUN	WOA	GTO	AVOA	EO	GWO	RSA	SMA	DE
Unimodal											
F90	1.21×10^{-12}	1.20×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	NaN
Multimodal											
F91	3.36×10^{-8}	4.03×10^{-9}	2.42×10^{-7}	2.07×10^{-10}	8.53×10^{-10}	2.44×10^{-9}	8.08×10^{-9}	5.76×10^{-11}	2.10×10^{-11}	1.40×10^{-10}	2.68×10^{-10}
F92	1.72×10^{-12}	1.72×10^{-12}	1.72×10^{-12}	1.72×10^{-12}	1.72×10^{-12}	1.72×10^{-12}	3.37×10^{-12}	1.72×10^{-12}	1.72×10^{-12}	1.72×10^{-12}	1.00×10^0
F93	3.32×10^{-6}	7.59×10^{-7}	3.16×10^{-10}	3.82×10^{-10}	8.86×10^{-10}	5.49×10^{-11}	6.63×10^{-1}	9.05×10^{-2}	3.02×10^{-11}	1.41×10^{-9}	3.69×10^{-11}
F94	3.44×10^{-10}	7.88×10^{-12}	7.88×10^{-12}	7.88×10^{-12}	7.88×10^{-12}	7.88×10^{-12}	1.20×10^{-9}	1.08×10^{-11}	7.88×10^{-12}	2.50×10^{-11}	4.22×10^{-5}

Table 15. Cont.

F	SSA	GBO	RUN	WOA	GTO	AVOA	EO	GWO	RSA	SMA	DE
Hybrid											
F95	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	8.99×10^{-11}
F96	2.99×10^{-11}	2.99×10^{-11}	2.99×10^{-11}	2.99×10^{-11}	2.99×10^{-11}	2.99×10^{-11}	4.03×10^{-11}	2.99×10^{-11}	2.99×10^{-11}	3.65×10^{-11}	1.15×10^{-2}
F97	4.50×10^{-11}	1.96×10^{-10}	3.34×10^{-11}	3.34×10^{-11}	8.15×10^{-11}	3.34×10^{-11}	2.87×10^{-10}	4.50×10^{-11}	3.02×10^{-11}	2.61×10^{-10}	1.30×10^{-1}
Composition											
F98	1.21×10^{-12}	1.61×10^{-1}	1.21×10^{-12}	1.21×10^{-12}	2.16×10^{-2}	4.65×10^{-8}	2.15×10^{-2}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	3.34×10^{-1}
F99	1.08×10^{-2}	8.15×10^{-11}	4.62×10^{-10}	3.02×10^{-11}	5.97×10^{-9}	4.62×10^{-10}	5.69×10^{-1}	5.46×10^{-6}	3.02×10^{-11}	2.64×10^{-1}	6.95×10^{-1}
F100	1.85×10^{-1}	2.36×10^{-1}	6.44×10^{-1}	9.41×10^{-10}	1.37×10^{-2}	5.00×10^{-1}	2.51×10^{-6}	7.91×10^{-9}	9.31×10^{-12}	8.02×10^{-1}	6.47×10^{-3}
F101	1.59×10^{-3}	3.62×10^{-6}	4.19×10^{-4}	5.38×10^{-11}	4.59×10^{-5}	6.66×10^{-7}	2.86×10^{-3}	3.80×10^{-5}	2.95×10^{-11}	4.29×10^{-1}	4.41×10^{-2}

4.8. The Overall Effectiveness of the Proposed Algorithm

In the previous sections, LSO has been separately assessed using five mathematical benchmarks: CEC2005, CEC2014, CEC2017, CEC2020, and CEC2022, and compared with twenty-two well-established metaheuristic algorithms, but the overall performance of LSO over all benchmarks has to be elaborated. Therefore, this section is presented to compare the overall performance of LSO and the other algorithms over all the test functions of each benchmark and all benchmarks. The average rank values and SD values of each benchmark are computed and reported in Table 16. This table also indicates the overall effectiveness of the proposed algorithm and other rival algorithms using an additional metric known as overall effectiveness (OE) and computed according to the following formula [127]:

$$OE (\%) = \frac{N' - L_i}{N'} \quad (29)$$

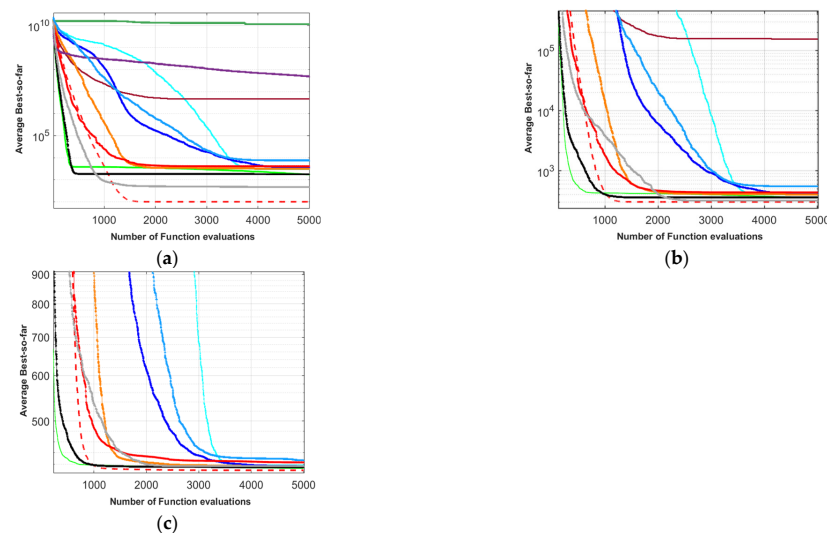
where N' denotes the total number of test functions, and L_i denotes the number of test functions in which the i -th algorithm is a loser. Inspecting this table reveals that the proposed algorithm could be superior in terms of SD, rank, and OE for four challengeable benchmarks, and competitive regarding rank and superior regarding SD and OE for the remaining benchmark. The average of the rank values, OE values, and SD values of each algorithm across all benchmarks are computed and reported in the last rows of Table 16, respectively, to measure the overall effectiveness of each algorithm across all benchmarks. According to those rows, LSO ranks first for all indicators, with a significant difference from the nearest well-performed algorithm. The LSO's strong performance due to the variation of the search process enables the algorithm to have a strong exploration and exploitation operator during the optimization process to help aid in preserving the population diversity, avoiding being stuck in local optima, accelerating the convergence towards the best-so-far solution. It is worth noting that both the inverse incomplete gamma function and the inverse random number are capable of preserving population diversity as well as avoiding becoming stuck in local minima due to their ability to generate significant numbers that aid in jumping the current solution to far away regions within the search space throughout the optimization process. On the other hand, three different scattering stages provide a variation in the exploitation operator to the LSO for rapidly reaching the near-optimal solution for various optimization problems of varying difficulty.

Table 16. Overall effectiveness over five investigated benchmarks (The bolded and red value is the best overall).

F	Index	LSO	SSA	GBO	RUN	WOA	GTO	AVOA	EO	GWO	RSA	SMA	DE
CEC2005	Rank	2.00	7.10	3.00	3.60	4.90	1.70	1.60	4.50	5.50	6.55	2.75	7.60
	SD	0.10	646.25	0.63	0.43	7101.71	0.14	0.18	0.43	0.59	0.29	0.24	5749.12
	$O \times 10$ (%)	85.00	25.00	50.00	50.00	40.00	70.00	85.00	30.00	25.00	45.00	65.00	20.00
CEC2014	Rank	1.66	7.28	5.45	6.55	9.72	7.07	7.48	4.79	8.41	11.93	6.21	4.45
	SD	22.57	17,917.37	10,668.42	22,738.20	256,429.26	21,129.59	17,868.18	2636.00	9.75×10^6	6.93×10^7	2470.27	6338.28
	$O \times 10$ (%)	79.31	0.00	3.45	6.90	0.00	0.00	0.00	6.90	0.00	0.00	6.90	6.90
CEC2017	Rank	1.11	6.43	5.57	7.39	11.21	5.46	8.50	4.43	8.11	13.00	6.68	3.32
	SD	32.47	52,413.76	40,719.95	14,459.54	446,168.54	103,439.53	26,051.38	17,788.92	4.44×10^6	1.47×10^8	17,155.85	10,876.18
	$O \times 10$ (%)	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.45
CEC2020	Rank	1.78	6.44	4.44	6.11	11.11	5.67	8.67	3.78	8.33	12.22	7.22	4.44
	SD	38.24	1161.15	318.84	1012.58	264,287.33	335.90	1482.46	518.82	1.60×10^7	4.15×10^8	1501.58	267.78
	$O \times 10$ (%)	90.00	0.00	10.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
CEC2022	Rank	1.64	6.00	4.73	7.00	11.45	6.00	8.45	5.09	8.91	12.82	5.36	3.64
	SD	11.73	172.93	128.52	119.23	782.34	32.96	173.73	184.05	420.22	2.01×10^7	206.53	17.99
	$O \times 10$ (%)	75.00	0.00	0.00	0.00	0.00	8.33	0.00	0.00	0.00	0.00	16.67	16.67
Avg. Rank:		1.64	6.65	4.64	6.13	9.68	5.18	6.94	4.52	7.85	11.30	5.64	4.69
Avg. SD:		21.02	14,462.29	10,367.27	7666.00	194,953.84	24,987.62	9115.19	4225.64	6.04×10^6	1.30×10^8	4266.89	4649.87
Avg.OE (%):		85.86	5.00	12.69	11.38	8.00	15.67	17.00	7.38	5.00	9.00	17.71	9.40

4.9. Convergence Curve

Figure 17 compares the convergence rates of LSO and competing algorithms to show how they differ in terms of reaching the near-optimal solution in less number function evaluations. This figure illustrates that all LSO convergence curves exhibit an accelerated reducing pattern within the various stages of the optimization process for four families of test functions (unimodal, multimodal, composition, and hybrid). The LSO optimizer is significantly faster than any of the other competing algorithms, as shown by the convergence curves in Figure 17. Exploration and exploitation operators of LSO can work together in harmony, which prevents stagnation in local minima and speeds up convergence in the right direction to the most promising regions.

**Figure 17.** Cont.

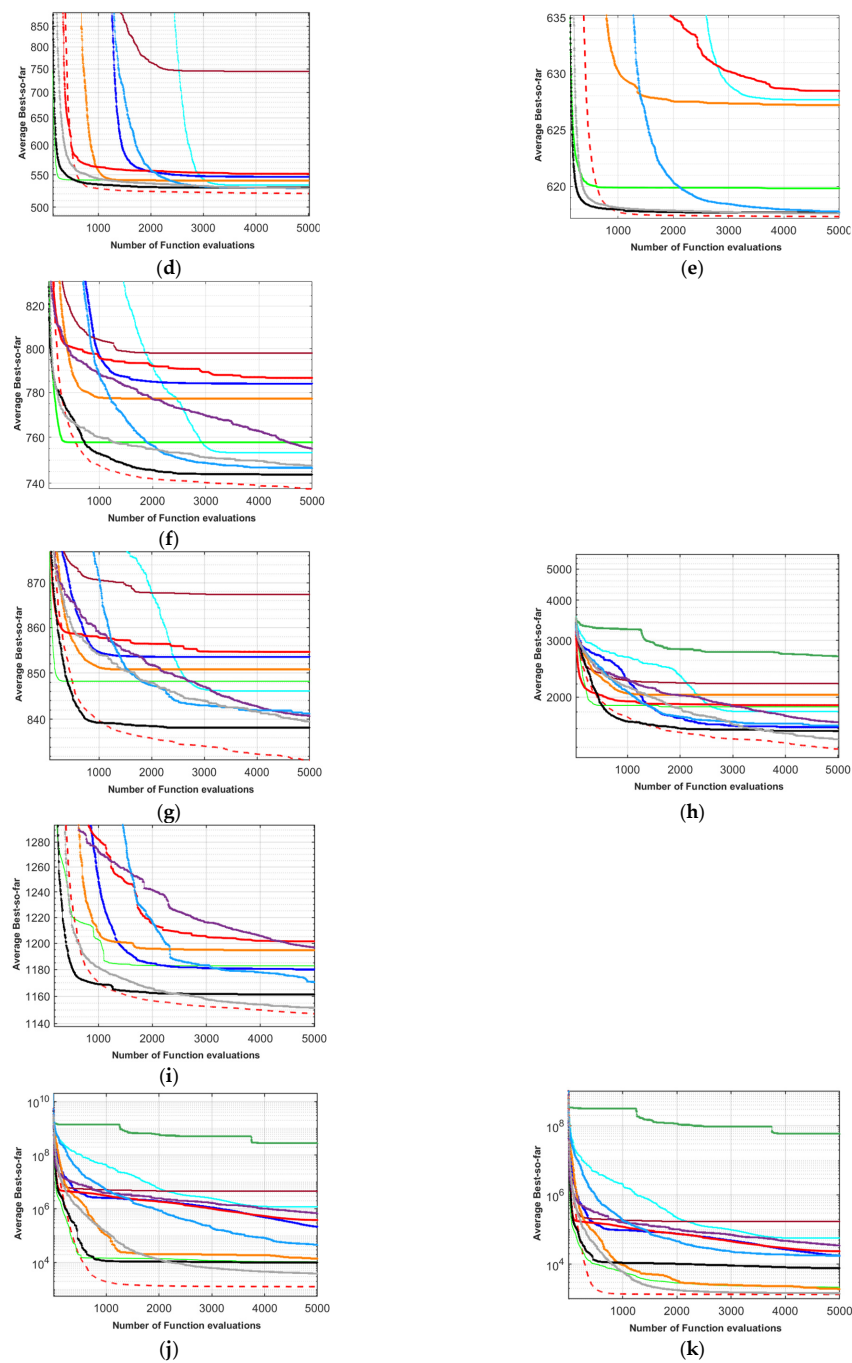


Figure 17. Cont.

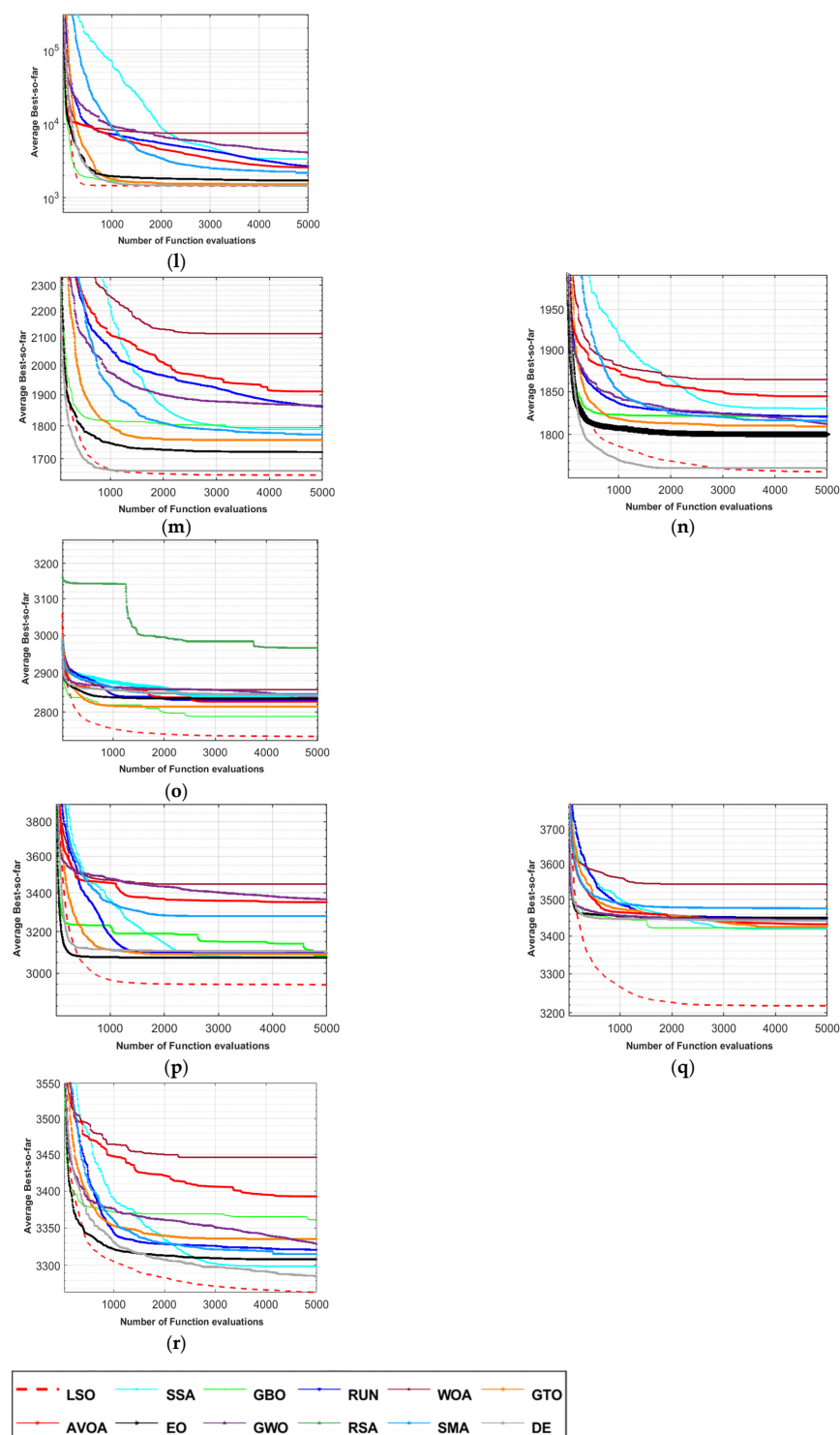


Figure 17. Depiction of averaged convergence speed among algorithms on some test functions. (a) F51 (Unimodal), (b) F52 (Unimodal), (c) F53 (Multimodal), (d) F54 (Multimodal), (e) F55 (Multimodal), (f) F56 (Multimodal), (g) F57 (Multimodal), (h) F59 (Multimodal), (i) F60 (Multimodal), (j) F61 (Multimodal), (k) F62 (Hybrid), (l) F63 (Hybrid), (m) F65 (Hybrid), (n) F64 (Hybrid), (o) F73 (Composition), (p) F75 (Composition), (q) F77 (Composition), and (r) F78 (Composition).

4.10. Qualitative Analysis

The following metrics are used to evaluate the LSO performance during optimization: diversity, convergence curve, average fitness value, trajectory in the first dimension, and search history. The diversity metric shows how far apart an individual is on average

from other individuals in the population; the convergence curve depicts the best-fitting value that was obtained within each iteration; the average fitness value represents the average of all individuals' fitness values throughout each iteration; the trajectory curve shows how a solution's first dimension changes over time as it progresses through the optimization process; and search history shows how a solution's position changed during the optimization process.

The diversity metric shown in the second column of Figure 18 is computed by summing the difference mean between the positions of each two solutions in the population according to the following formula:

$$\aleph = \sum_{i=1}^N \sum_{j=i+1}^{N-1} \frac{(\sum_{k=0}^d |x_{i,k} - x_{j,k}|)}{d} \quad (30)$$

where d indicates the number of dimensions, N stands for the population size, $x_{i,k}$ indicate the k th dimension in the i th and j th solutions such that $j > i$. Observing Figure 18 shows that the diversity metric of LSO is decreasing over time, indicating that LSO's optimization process is gradually shifting from exploration to exploitation. LSO's performance initially started to explore most regions in the search space to avoid stagnation into local minima and then shifted gradually to the exploitation operator to reduce diversity quickly during the second half of the optimization process to accelerate the convergence toward the most promising region discovered thus far.

The LSO convergence curves show an accelerated reducing pattern on a variety of test functions during the latter half of the optimization process when population diversity is reduced. The exploratory phase is largely transformed into the exploitative phase, as illustrated in the third column of Figure 18. At the beginning of the optimization process, LSO convergence is slow to avoid becoming stuck in local minima. Then, it is highly improved in the second half of the optimization process.

The depiction of average fitness history in Figure 18 shows that LSO's competitiveness has decreased over time due to all solutions focusing on exploiting the regions around the best-so-far solution, and as a result, the fitness values of all solutions are nearly moved towards the same region which involves the best-so-far solution. Figure 18 also depicts the trajectory of LSO's search for the optimal position of the first dimension as it gradually explores all aspects of the search space, as depicted in Figure 18. Because of the need to find a better solution in a short period of time, the exploratory approach is being replaced by an exploitative approach that restricts the scope of the search to a single aspect of it. As can be seen from the LSO's trajectory curve, the optimization process begins with an exploratory trend before moving to an exploitation trend in search of better outcomes before coming to an end.

In the final column of Figure 18, the history of LSO positions is depicted. The search history is investigated by depicting the search core followed by LSO's solutions within the whole optimization process for the first two dimensions: X_1 and X_2 of an optimization problem. The same pattern is followed for the other dimensions. As can be seen in this preceding column, LSO does not follow a consistent pattern for all test functions. Consider F21 as an example: to find an optimal solution for this problem, LSO first explores the entire search space before narrowing its focus to the range 0–50. The search history graph shows that LSO's performance is more dispersed for the multimodal and composition test functions, while its performance for the unimodal test function is more concentrated around the optimum points.

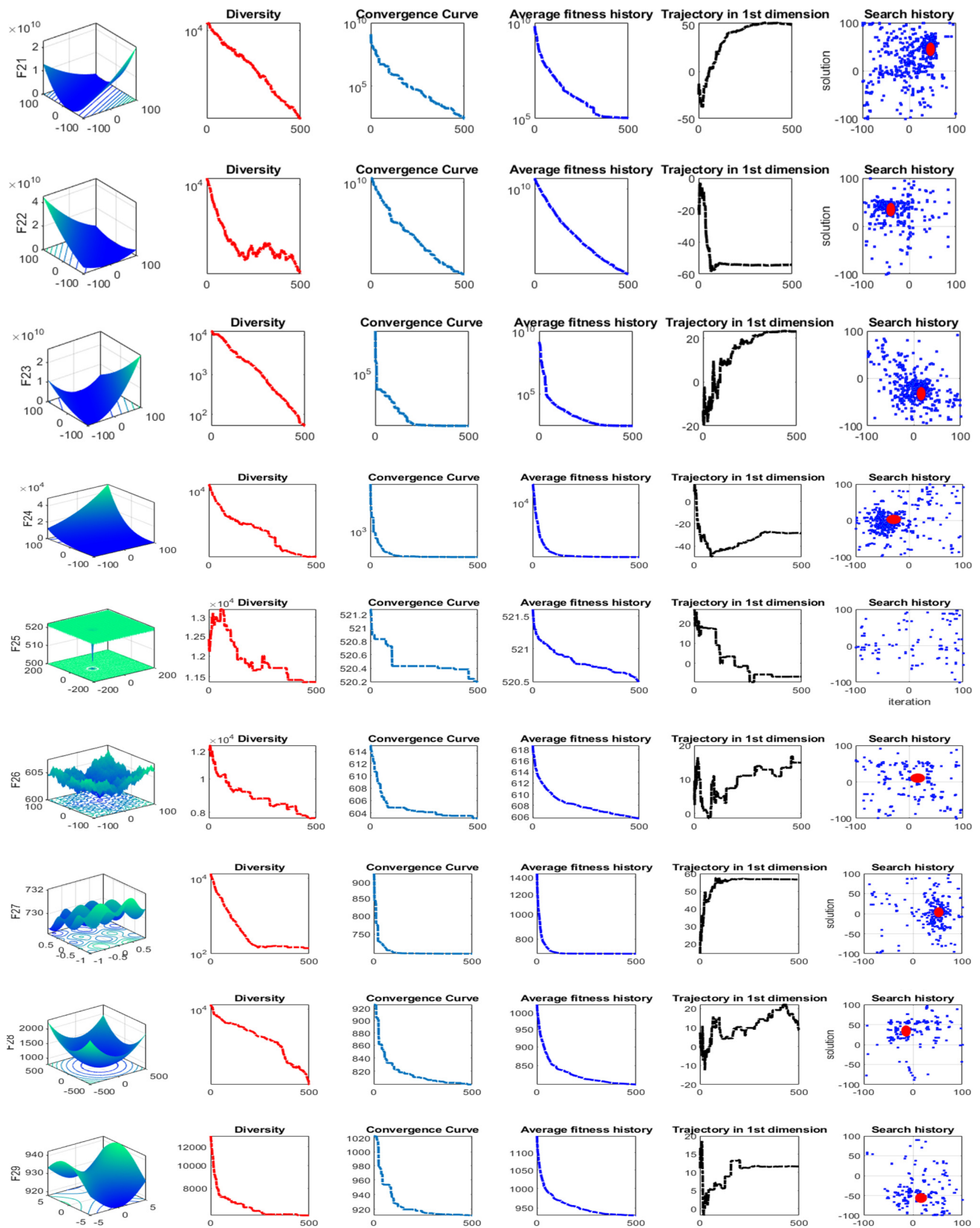


Figure 18. Diversity, convergence curve, average fitness history, trajectory, and search history.

4.11. Computational Cost

The average computational cost consumed by each algorithm on the investigated test functions is shown in Figure 19. The graph shows that the CPU time for all algorithms is nearly the same, except RSA and SMA, which take a long time, and WOA, which takes less than half the time required by the rest. LSO is thus far superior in terms of the convergence speed and the quality of the obtained outcomes, with a negligible difference in CPU time.

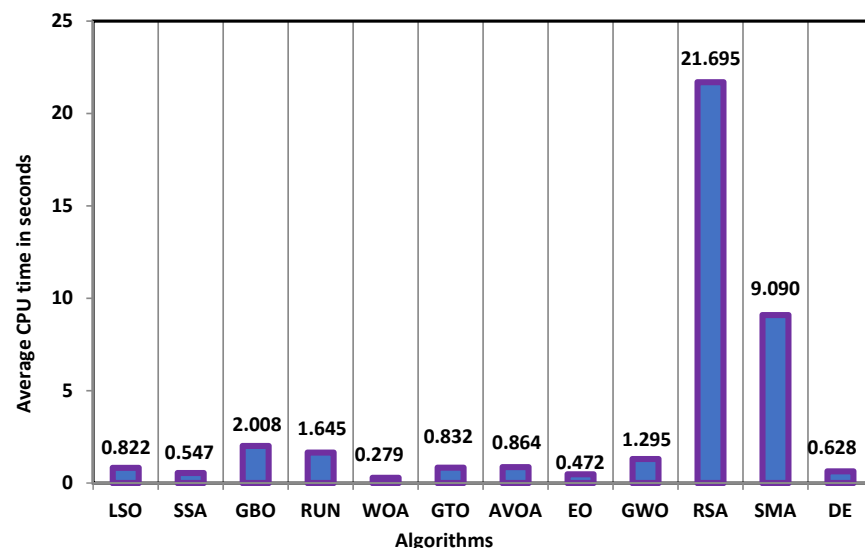


Figure 19. Comparison among algorithms in terms of CPU time.

5. LSO for Engineering Design Problems

In section, LSO is applied to solve three constrained engineering benchmarks, including Tension/Compression Spring Design Optimization Problem, Welded Beam Design Problem, and Pressure Vessel Design Problem. The best values found by LSO during 25 runs are compared with many optimization algorithms. For compared the algorithms' parameters settings, all parameters are left as the defaults suggested by the authors. The parameters of LSO are kept as mentioned in Table 5, except for P_s , which substantially affect the performance of LSO based on the nature of the solved problems. Therefore, an extensive experiment has been done under various values for this parameter, and the obtained outcomes are depicted in Figure 20. This figure shows that the performance of LSO is maximized when $P_s = 0.6$. In addition to all rival algorithms used in the previous comparisons, five recently-published metaheuristic algorithms known as political optimizer (PO) [45], continuous-state cellular automata algorithm (CCAA) [128], snake optimizer (SO) [91], beluga whale optimization (BWO), [96] and driving training-based optimization (DTBO) [50] are added in the next experiments to further show the superiority of LSO when tackling the real-world optimization problems, such as engineering design problems. Additionally, LSO is compared to some of the state-of-the-art optimizers proposed for each constrained engineering benchmark according to the cited results.

Engineering design problems are characterized by many different constraints. In order to handle this type of constraint, we employ penalty-based constraint handling techniques with LSO. There are several methods for handling constraints of optimization problems based on the penalty function. In this work, we choose to implement the Death Penalty method (the rejection of infeasible solutions method) [129], in which the infeasible solutions are rejected and regenerated. So, the infeasible solution is automatically omitted from the candidate solutions. The main advantage of the Death Penalty method is its simple implementation and low computational complexity.

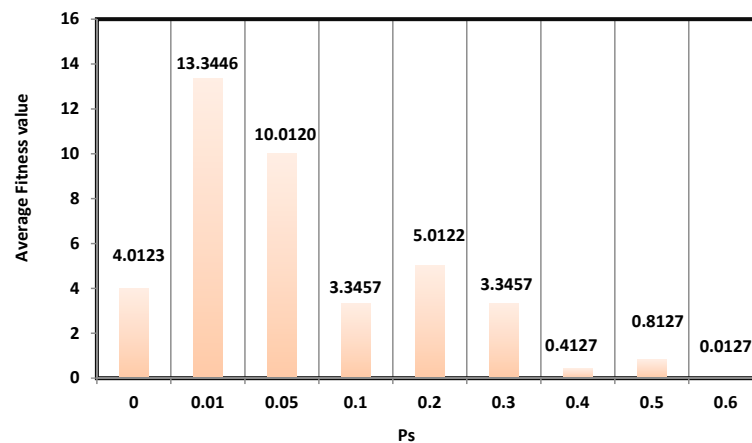


Figure 20. Tuning the parameter P_s over tension spring design.

5.1. Tension/Compression Spring Design Optimization Problem

The main objective of the tension/compression spring design optimization problem is to find the minimum volume $f(X)$ of a coil spring under compression undergoing constraints of minimum deflection, shear stress, surge frequency, and limits on outside diameter and design variables (See Figure 21a). Mathematically, the problem can be formulated as [130]:

$$\begin{aligned}
 \min f(X) &= (x_3 + 2)x_2x_1^2 \\
 \text{s.t. } g_1(X) &= 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0 \\
 g_2(X) &= \frac{x_2(4x_2 - x_1)}{x_1^3(12566x_2 - x_1)} + \frac{1}{5108x_1^2} - 1 \leq 0 \\
 g_3(X) &= 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0 \\
 g_4(X) &= \frac{2(x_1 + x_2)}{3} - 1 \leq 0 \\
 0.05 &\leq x_1 \leq 2, \quad 0.25 \leq x_2 \leq 1.3, \quad 2 \leq x_3 \leq 15,
 \end{aligned} \tag{31}$$

where x_1 is the wire diameter, x_2 is the mean coil diameter, and x_3 is the length or number of coils.

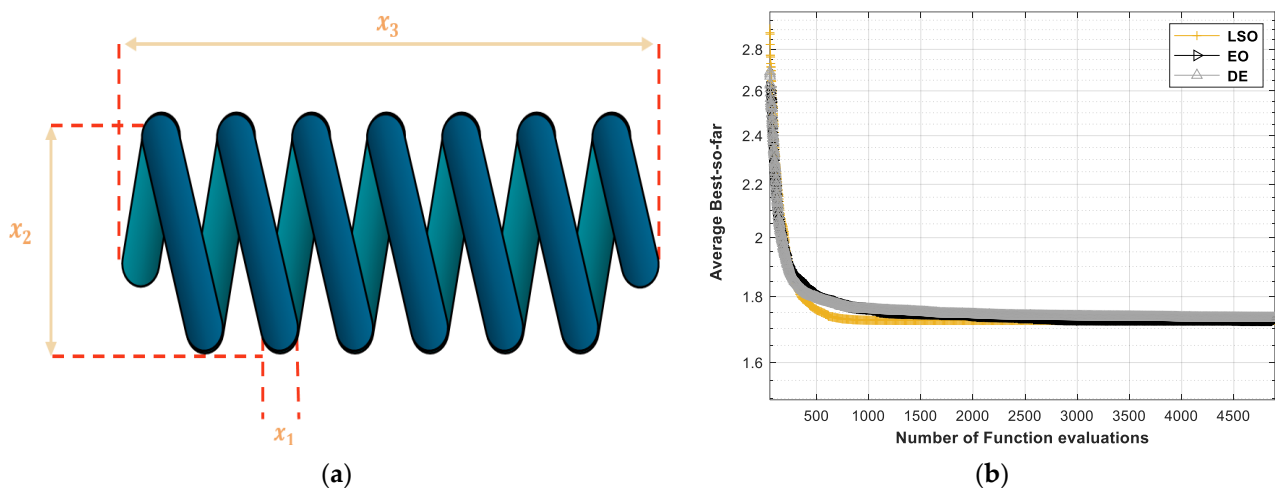


Figure 21. Tension/compression spring design problem. (a) Structure. (b) Convergence curve.

LSO is compared with 14 additional algorithms (mathematical techniques or meta-heuristics) selected from various literature including evolution strategies (ES) [131], gravitational search algorithm (GSA) [55], Tabu search (TS) [132], swarm strategy [133], unified particle swarm optimization (UPSO) [134], cultural algorithms (CA) [132], two-point adaptive nonlinear approximations-3 (TANA-3) [135], particle swarm optimization (PSO) [136], ant colony optimization (ACO) [137], genetic algorithm (GA) [138], quantum evolutionary algorithm (QEA) [137], ray optimization (RO) [67], probability collectives (PC) [139], social interaction genetic algorithm (SIGA) [140], and parallel genetic algorithm with social interaction (PSIGA) [138]. Table 17 shows the results obtained by LSO for tension/compression spring design optimization problem. As shown, LSO is better able to reach minimum values than others. Figure 21b shows the convergence speed of LSO.

Table 17. Comparison of the results for tension/compression spring design optimization problem.

No.	Algorithm	x_1	x_2	x_3	$f_{min}(x)$
1	LSO	5.1689×10^{-2}	3.5671×10^{-1}	1.1290×10^1	1.2665233×10^{-2}
2	SSA	5.0000×10^{-2}	3.1737×10^{-1}	1.4051×10^1	1.2735631×10^{-2}
3	GBO	5.3274×10^{-2}	3.9605×10^{-1}	9.3074×10^0	1.2709931×10^{-2}
4	RUN	5.1960×10^{-2}	3.6328×10^{-1}	1.0914×10^1	1.2666568×10^{-2}
5	WOA	5.2797×10^{-2}	3.8396×10^{-1}	9.8542×10^0	1.2687243×10^{-2}
6	GTO	5.2437×10^{-2}	3.7499×10^{-1}	1.0293×10^1	1.2675330×10^{-2}
7	AVOA	5.1702×10^{-2}	3.5704×10^{-1}	1.1270×10^1	1.2665236×10^{-2}
8	EO	5.3212×10^{-2}	3.9449×10^{-1}	9.3755×10^0	1.2706550×10^{-2}
9	GWO	5.2770×10^{-2}	3.8328×10^{-1}	9.8863×10^0	1.2686160×10^{-2}
10	RSA	5.1387×10^{-2}	3.4946×10^{-1}	1.1734×10^1	1.2673204×10^{-2}
11	SMA	5.0000×10^{-2}	3.1050×10^{-1}	1.5000×10^1	1.3196460×10^{-2}
12	DE	5.2354×10^{-2}	3.7292×10^{-1}	1.0399×10^1	1.2673207×10^{-2}
13	PO	5.1778×10^{-2}	3.5884×10^{-1}	1.1166×10^1	$1.26665924 \times 10^{-2}$
14	SO	5.1403×10^{-2}	3.4988×10^{-1}	1.1701×10^1	$1.26667301 \times 10^{-2}$
15	BWO	5.4611×10^{-2}	4.2818×10^{-1}	9.0696×10^0	$1.41360583 \times 10^{-2}$
16	DTBO	5.3059×10^{-2}	3.8966×10^{-1}	9.6233×10^0	$1.27507490 \times 10^{-2}$
17	CCAA	5.2012×10^{-2}	3.6453×10^{-1}	1.0846×10^1	$1.26674032 \times 10^{-2}$
18	RO	5.13700×10^{-2}	3.49096×10^{-1}	1.17628×10^1	1.2678800×10^{-2}
19	ES	5.1989×10^{-2}	3.6397×10^{-1}	1.08905×10^1	1.2681×10^{-2}
20	GSA	5.02760×10^{-2}	3.2368×10^{-1}	1.352541×10^1	1.27022×10^{-2}
21	TS	N/A	N/A	N/A	1.2935×10^{-2}
22	Swarm strategy	5.0417×10^{-2}	3.2153×10^{-1}	1.3980×10^{-1}	1.3060×10^{-2}
23	UPSO	N/A	N/A	N/A	1.31×10^{-2}
24	CA	N/A	N/A	N/A	1.2867×10^{-2}
25	TANA-3	5.8400×10^{-2}	5.4170×10^{-1}	5.2745×10^0	1.3400×10^{-2}
26	PSO	N/A	N/A	N/A	1.2857×10^{-2}
27	ACO	N/A	N/A	N/A	1.3223×10^{-2}
28	GA	5.8231×10^{-2}	5.2106×10^{-1}	5.8845×10	1.3931×10^{-2}
29	QEA	N/A	N/A	N/A	1.2928×10^{-2}
30	PC	5.06×10^{-2}	3.28×10^{-1}	1.41×10^{-1}	1.35×10^{-2}
31	SIGA	N/A	N/A	N/A	1.3076×10^{-2}
32	PSIGA	N/A	N/A	N/A	1.2864×10^{-2}

5.2. Welded Beam Design Problem

The problem of designing welded beams can be defined as finding the feasible dimensions of a welded beam x_1 , x_2 , x_3 , and x_4 (which are the thickness of weld, length of the clamped bar, the height of the bar, and thickness of the bar, respectively) that minimize the total manufacturing cost $f(X)$ subject to a set of constraints. Figure 22a shows a representa-

tion of the weld beam design problem. Mathematically, the problem can be formulated as the following [130]:

$$\begin{aligned}
 \min f(X) &= 1.10471x_1^2x_2 + 0.04811x_3x_4(L + x_2) \\
 g_1(X) &= \tau(X) - \tau_{max} \leq 0 \\
 &\quad s.t. \\
 g_2(X) &= \sigma(X) - \sigma_{max} \leq 0 \\
 g_3(X) &= x_1 - x_4 \leq 0 \\
 g_4(X) &= 0.10471x_1^2 - 0.04811x_3x_4(L + x_2) - 5 \leq 0 \\
 g_5(X) &= 0.125 - x_1 \leq 0 \\
 g_6(X) &= \delta(X) - \delta_{max} \leq 0 \\
 g_7(X) &= P - P_C(X) \leq 0 \tag{32} \\
 \tau(X) &= \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}, \quad \tau' = \frac{P}{\sqrt{2}x_1x_2}, \quad \tau'' = \frac{MR}{J} \\
 M &= P(L + \frac{x_2}{2}), \quad R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}, \\
 J &= 2\left(\sqrt{2}x_1x_2\left(\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right)\right) \\
 \sigma(X) &= \frac{6PL}{x_3^2x_4}, \quad \delta(X) = \frac{4PL^3}{Ex_3^3x_4}, \\
 P_C(X) &= \frac{4.013E\sqrt{\left(\frac{x_2^2x_4^6}{36}\right)}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right) \\
 P &= 6000 \text{ lb}, \quad L = 14 \text{ in}, \quad E = 30 \times 10^6 \text{ psi}, \quad G = 12 \times 10^6 \text{ psi} \\
 \tau_{max} &= 13600 \text{ psi}, \quad \sigma_{max} = 30000 \text{ psi}, \quad \delta_{max} = 0.25 \text{ in} \\
 0.1 &\leq x_1 \leq 2, \quad 0.1 \leq x_2 \leq 10, \quad 0.1 \leq x_3 \leq 10, \quad 0.1 \leq x_4 \leq 2,
 \end{aligned}$$

where τ is shear stress, σ is the bending stress, P_C is the buckling load, and δ is the end deflection.

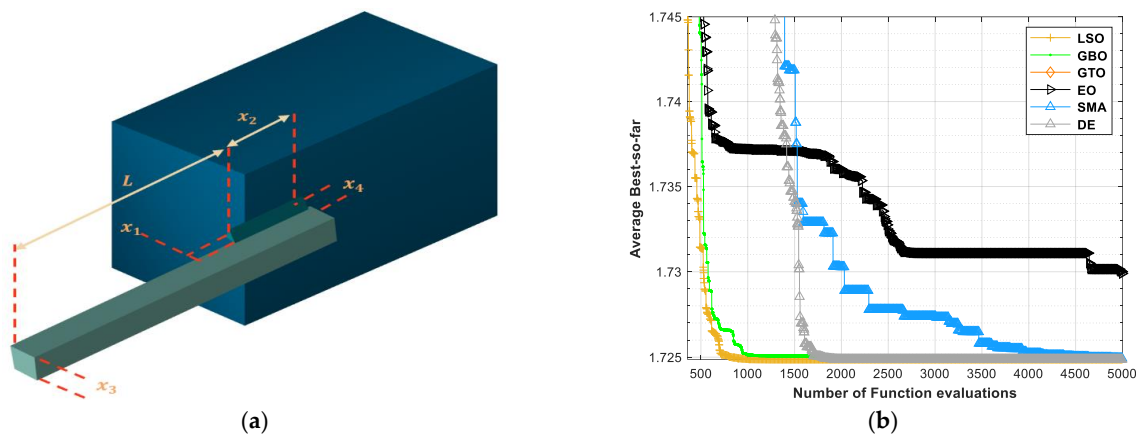


Figure 22. Weld beam design problem. (a) Structure. (b) Convergence curve.

The proposed algorithm is compared with nine additional algorithms from the literature including hybrid and improved ones, which are RO [67], WOA [41], HS [141], hybrid charged system search and particle swarm optimization (PSO) algorithms I (CSS&PSO I) [136], hybrid charged system search and particle swarm optimization (PSO) algorithms II (CSS&PSO II) [136], particle swarm optimization algorithm with struggle selection (PSOStr) [140], RO [67], firefly algorithm (FA) [142], differential evolution with dynamic stochastic selection (DE) [143], and hybrid artificial immune system and genetic algorithm (AIS-GA). As shown in Table 18, on one side, LSO has a competitive result comparing to GBO, GTO, EO, and DE, and superior in terms of the convergence speed as shown in Figure 22b. On the other side, LSO is superior to all the other optimizers.

Table 18. Comparison of the results for weld beam design problem.

No.	Algorithm	x_1	x_2	x_3	x_4	$f_{min}(x)$
1	LSO	2.0572×10^{-1}	3.4707×10^0	9.0366×10^0	2.0573×10^{-1}	1.7248658
2	SSA	2.0548×10^{-1}	3.4757×10^0	9.0369×10^0	2.0573×10^{-1}	1.7252086
3	GBO	2.0572×10^{-1}	3.4707×10^0	9.0366×10^0	2.0573×10^{-1}	1.7248658
4	RUN	2.0572×10^{-1}	3.4708×10^0	9.0366×10^0	2.0573×10^{-1}	1.7248746
5	WOA	1.9721×10^{-1}	3.6965×10^0	9.0151×10^0	2.0671×10^{-1}	1.7454019
6	GTO	2.0572×10^{-1}	3.4707×10^0	9.0366×10^0	2.0573×10^{-1}	1.7248658
7	AVOA	2.0567×10^{-1}	3.4718×10^0	9.0366×10^0	2.0573×10^{-1}	1.7249388
8	EO	2.0572×10^{-1}	3.4707×10^0	9.0366×10^0	2.0573×10^{-1}	1.7248658
9	GWO	2.0557×10^{-1}	3.4741×10^0	9.0376×10^0	2.0573×10^{-1}	1.7253039
10	RSA	2.0615×10^{-1}	3.4694×10^0	9.0775×10^0	2.2698×10^{-1}	1.8945096
11	SMA	2.0572×10^{-1}	3.4707×10^0	9.0366×10^0	2.0573×10^{-1}	1.7248689
12	DE	2.0572×10^{-1}	3.4707×10^0	9.0366×10^0	2.0573×10^{-1}	1.7248658
13	PO	2.0572×10^{-1}	3.4707×10^0	9.0366×10^0	2.0573×10^{-1}	1.72486608
14	SO	2.0572×10^{-1}	3.4707×10^0	9.0366×10^0	2.0573×10^{-1}	1.7248658
15	BWO	1.8639×10^{-1}	3.9876×10^0	8.9766×10^0	2.1300×10^{-1}	1.8076732
16	DTBO	2.0571×10^{-1}	3.4709×10^0	9.0368×10^0	2.0574×10^{-1}	1.7249588
17	CCAA	2.0572×10^{-1}	3.4707×10^0	9.0367×10^0	2.0573×10^{-1}	1.72487987
18	RO	2.03687×10^{-1}	3.52847×10^0	9.00423×10^0	2.07241×10^{-1}	1.735344
19	RO	2.03687×10^{-1}	3.52847×10^0	9.00423×10^0	2.072410×10^{-1}	1.735344×10^0
20	WOA	2.05396×10^{-1}	3.48429×10^0	9.03743×10^0	2.06276×10^{-1}	1.73050×10^0
21	HS	2.4420×10^{-1}	6.2231×10^0	8.29150×10^0	2.4430×10^{-1}	2.38070×10^0
22	CSS&PSO I	2.0639×10^{-1}	3.4236×10^0	9.1241×10^0	2.0531×10^{-1}	1.7314×10^0
23	CSS&PSO II	2.0546×10^{-1}	3.4800×10^0	9.05401×10^0	2.0578×10^{-1}	1.72910×10^0
24	PSOStr	2.0150×10^{-1}	3.5620×10^0	9.0414×10^0	2.0571×10^{-1}	1.73118×10^0
25	FA	2.015×10^{-1}	3.5620×10^0	9.0414×10^0	2.0570×10^{-1}	1.73121×10^0
26	DE	2.444×10^{-1}	6.2175×10^0	8.2915×10^0	2.4440×10^{-1}	2.3810×10^0
27	AIS-GA	2.444×10^{-1}	6.2183×10^0	8.2912×10^0	2.444×10^{-1}	2.3812×10^0

5.3. Pressure Vessel Design Problem

The problem of pressure vessel design can be described as minimizing the total fabrication cost of a cylindrical pressure vessel with the consideration of optimization constraints. The mathematical formulation of the problem can be expressed as [41]:

$$\begin{aligned}
 \min f(X) &= 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \\
 \text{s.t.} \quad &g_1(X) = -x_1 + 0.0193x_3 \leq 0 \\
 &g_2(X) = -x_3 + 0.00954x_3 \leq 0 \\
 &g_3(X) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0 \\
 &g_4(X) = x_4 - 240 \leq 0 \\
 &0 \leq x_1, x_2 \leq 99, 10 \leq x_3, x_4 \leq 200,
 \end{aligned} \tag{33}$$

where x_1 is the shell thickness, x_2 is the head thickness, x_3 is the inner radius, and x_4 is the cylindrical section length (without the head), as shown in Figure 23a.

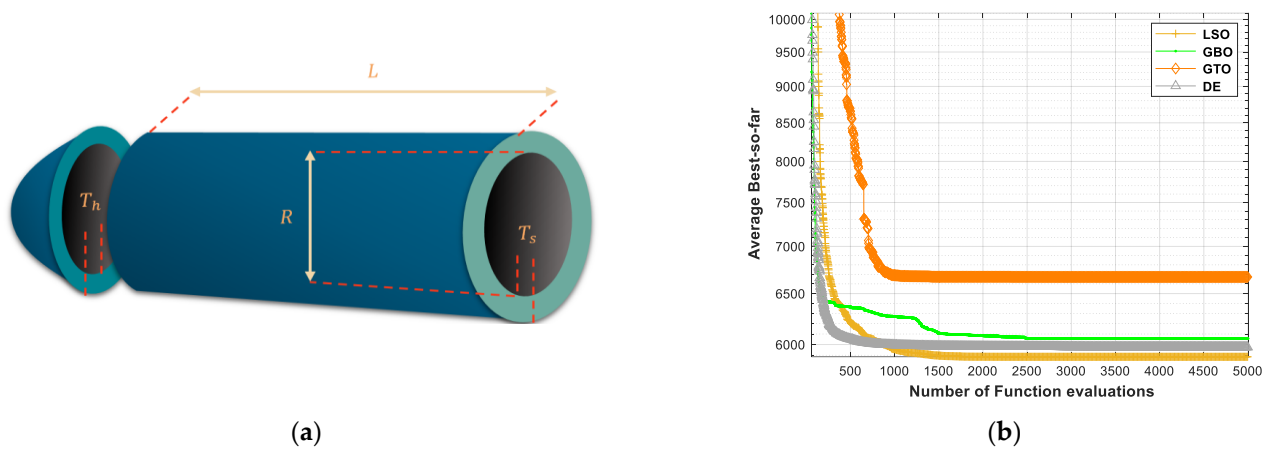


Figure 23. Pressure vessel design problem. (a) Structure. (b) Convergence curve.

For the problem of pressure vessel design, the proposed algorithm is compared with 16 metaheuristics and mathematical methods, including Harris Hawks optimization (HHO) [22], grey wolf optimizer (GWO) [122], hybrid particle swarm optimization with a feasibility-based rule (HPSO) [144], Gaussian quantum-behaved particle swarm optimization (G-QPSO) [145], water evaporation optimization (WEO) [146], bat algorithm (BA) [147], MFO [148], charged system search (CSS) [56], multimembered evolution strategy (ESs) [149], genetic algorithm for design and optimization of composite laminates (BIANCA) [150], modified differential evolution (MDDE) [151], differential evolution with level comparison (DELIC) [152], WOA [41], niched-pareto genetic algorithm (NPGA) [153], Lagrangian multiplier [41], and branch-bound [41]. As observed from Table 19, LSO significantly reaches the best results for the given problem compared to other algorithms except for DE and GTO, which have competitive performance for the best result, but LSO is superior for the average convergence speed as shown in Figure 23b.

Table 19. Comparison of the results for pressure vessel design optimization problem.

No.	Algorithm	x_1	x_2	x_3	x_4	$f_{min}(x)$
1	LSO	7.7818×10^{-1}	3.8466×10^{-1}	4.0320×10^1	2.0000×10^2	5885.43417456
2	SSA	7.8272×10^{-1}	3.8690×10^{-1}	4.0555×10^1	1.9675×10^2	5893.25120195
3	GBO	7.7818×10^{-1}	3.8466×10^{-1}	4.0320×10^1	2.0000×10^2	5885.44199882
4	RUN	7.7824×10^{-1}	3.8471×10^{-1}	4.0323×10^1	1.9995×10^2	5885.60517789
5	WOA	1.0863×10^0	5.5859×10^{-1}	5.6156×10^1	5.5940×10^1	6779.76692510
6	GTO	7.7818×10^{-1}	3.8466×10^{-1}	4.0320×10^1	2.0000×10^2	5885.43417456
7	AVOA	7.7820×10^{-1}	3.8467×10^{-1}	4.0321×10^1	1.9998×10^2	5885.47556956
8	EO	8.2691×10^{-1}	4.0875×10^{-1}	4.2845×10^1	1.6760×10^2	5974.05971856
9	GWO	7.7854×10^{-1}	3.8509×10^{-1}	4.0332×10^1	1.9986×10^2	5888.33742107
10	RSA	1.2411×10^0	9.5886×10^{-1}	4.5644×10^1	1.3730×10^2	10,457.57367139
11	SMA	7.7818×10^{-1}	3.8466×10^{-1}	4.0320×10^1	2.0000×10^2	5885.43624519
12	DE	7.7818×10^{-1}	3.8466×10^{-1}	4.0320×10^1	2.0000×10^2	5885.43417456
13	PO	7.7818×10^{-1}	3.8466×10^{-1}	4.0320×10^1	2.0000×10^2	5885.43636261
14	SO	7.7818×10^{-1}	3.8470×10^{-1}	4.0320×10^1	2.0000×10^2	5885.54986448
15	BWO	7.9249×10^{-1}	3.9708×10^{-1}	4.0982×10^1	1.9550×10^2	6036.98073005
16	DTBO	8.7638×10^{-1}	4.5781×10^{-1}	4.5407×10^1	1.3954×10^2	6165.77458390
17	CCAA	7.7837×10^{-1}	3.8488×10^{-1}	4.0327×10^1	1.9989×10^2	5886.45565028
18	HHO	0.817583	0.40729	42.0917	176.7196	6000.46259
19	GWO	0.8125	0.4345	42.089181	176.758731	6051.5639
20	HPSO	0.8125	0.437500	42.0984	176.6366	6059.7143

Table 19. Cont.

No.	Algorithm	x_1	x_2	x_3	x_4	$f_{min}(x)$
21	G-QPSO	0.8125	0.437500	42.0984	176.6372	6059.7208
22	WEO	0.8125	0.437500	42.0984	176.6366	6059.71
23	BA	0.8125	0.437500	42.098445	176.63659	6059.7143
24	MFO	0.8125	0.4375	42.098445	176.636596	6059.7143
25	CSS	0.8125	0.4375	42.103624	176.572656	6059.0888
26	ESs	0.8125	0.4375	42.098087	176.640518	6059.7456
27	BIANCA	0.8125	0.4375	42.0968	176.658	6059.9384
28	MDDE	0.8125	0.4375	42.098446	176.636047	6059.70166
29	DELIC	0.8125	0.4375	42.0984456	176.6365958	6059.7143
30	WOA	0.8125	0.4375	42.0982699	176.638998	6059.7410
31	NPGA	0.8125	0.4375	42.0974	176.654	6059.9463
32	Lagrangian multiplier	1.125	0.625	58.291	43.69	7198.0428
33	Branch-bound	1.125	0.625	47.7	117.701	8129.1036

6. Conclusions

In this work, a novel LSO metaheuristic algorithm is introduced that is inspired by sunlight dispersion through a water droplet, causing the rainbow phenomenon. The proposed algorithm is tested on several selected benchmarks. For CEC2005 benchmarks, LSO significantly performs well, especially for fixed dimensional multi-model functions. This indicates that LSO has high exploration capabilities. In addition, the sensitivity analysis of LSO parameters shows that the selected values of the parameters are the best. Finally, for CEC2020, CEC2017, CEC2022, and CEC2014, LSO has a superior performance compared to several well-established and recently published metaheuristic algorithms like DE, WOA, SMA, EO, GWO, GTO, GBO, RSA, SSA, RUN, and AVOA, which have been selected in our comparison due to their stability and recent publication compared to some of the other optimization algorithms like MBO, EWA, EHO, MS, HGS, CPA, and HHO proposed for tackling the same benchmarks. This indicates that LSO has a good balance between exploration and exploitation. LSO has competitive results for engineering design problems compared to other algorithms, even for improved and hybrid metaheuristics. For future work, we suggest developing the binary and multi-objective versions of LSO. In addition, several enhancements can be proposed for LSO by using fuzzy controllers or chaotic maps for defining LSO controlling probabilities and the hybridization with other algorithms. Finally, we suggest using LSO for solving recent real-life optimization problems such as sensor allocation problems, smart management of the power grid, and smart routing of vehicles.

Author Contributions: Conceptualization, M.A.-B. and R.M.; methodology, M.A.-B., R.M. and K.M.S.; software, M.A.-B. and R.M.; validation, M.A.-B., R.M., R.K.C. and K.M.S.; formal analysis, M.A.-B., R.M., R.K.C. and K.M.S.; investigation, M.A.-B., R.M., R.K.C. and K.M.S.; data curation, M.A.-B., R.M., R.K.C. and K.M.S.; writing—original draft preparation, M.A.-B. and R.M.; writing—review and editing, M.A.-B., R.M., R.K.C. and K.M.S.; visualization, M.A.-B., R.M., R.K.C. and K.M.S.; supervision, M.A.-B. and R.K.C.; funding acquisition, R.K.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The code used in this paper can be obtained from this publicly accessible platform: <https://drive.matlab.com/sharing/633c724c-9b52-4145-beba-c9e49f9df42e> (accessed on 9 January 2022).

Acknowledgments: We would like to thank the Editor and anonymous reviewers for their thorough comments, which have helped us improve the manuscript—also, special thanks to Seyedali Mirjalili for his excellent suggestions and comments.

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

Nomenclature

Nomenclature of symbols used in this study

θ	Angle of reflection or refraction
$k \in \mathbb{R}$	Refractive index of a medium
$L_i (i = 0, \dots, 3)$	i^{th} refracted or reflected light ray
n_s	Normal line at a point s
$p \in [0, 1]$	Controlling probability of inner and outer reflection and refraction
$q \in [0, 1]$	Controlling probability of the first scattering phase
$z \in [0, 1]$	Controlling probability of the second scattering phase
$t \in \mathbb{N}$	Iteration number
$\vec{x}^0 \in \mathbb{R}$	Initial candidate solution
$N \in \mathbb{N}$	Population size
$d \in \mathbb{N}$	Problem dimension
$lb \in \mathbb{R}$	Lower bound of the search space
$ub \in \mathbb{R}$	Upper bound of the search space
$RV \in [0, 1]$	Vector of uniform random numbers
$\vec{x}_t \in \mathbb{R}$	Candidate solution at iteration t
$w \in [0, \infty]$	Scaling factor
$GI \in \mathbb{R}$	Scaling factor
$\epsilon \in \mathbb{R}$	Scaling factor
g_{inv}	Inverse incomplete gamma function

Appendix A

Table A1. Description of uni-modal benchmark functions.

ID	Benchmark	D	Domain	Global Opt.
F1	$f_1(x) = \sum_{i=1}^n x_i^2$	100	$[-100, 100]$	0
F2	$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i$	100	$[-10, 10]$	0
F3	$f_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	100	$[-100, 100]$	0
F4	$f_4(x) = \max_i \{ x_i , 1 \leq i \leq n\}$	100	$[-100, 100]$	0
F5	$f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	100	$[-30, 30]$	0

Table A2. Description of multi-modal benchmark functions.

ID	Benchmark	D	Domain	Global Opt.
F6	$f_6(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	100	$[-5.12, 5.12]$	0
F7	$f_7(x) = -20e^{(-0.2\sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2})} - e^{(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i))} + 20 + e$	100	$[-32, 32]$	0
F8	$f_8(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	100	$[-600, 600]$	0
F9		100	$[-50, 50]$	0
F10	$y_i = 1 + \frac{x_i+1}{4}, u(x_i, s, l, v) = \begin{cases} l(x_i - s)^v & x_i > s \\ 0 & -s < x_i < s \\ l(-x_i - s)^v & x_i < -s \end{cases}$ $f_{10}(x) = 0.1 \{ \sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] + \sum_{i=1}^n u(x_i, 5, 100, 4) \}$	100	$[-1.28, 1.28]$	0

Table A3. Fixed-dimension multi-modal benchmark.

ID	Benchmark	D	Domain	Global Opt.
F11	$f_{11}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right)^{-1}$	2	$[-65, 65]$	1
F12	$f_{12}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_i (b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	$[-5, 5]$	0.00030
F13	$f_{13}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]$	-1.0316
F14	$f_{14}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos x_1 + 10$	2	$[-5, 5]$	0.39789
F15	$f_{15}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$ $\times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	$[-2, 2]$	3
F16	$f_{16}(x) = - \sum_{i=1}^4 c_i \exp \left(- \sum_{j=1}^6 a_{ij} (x_j - p_{ij})^2 \right)$	3	$[1, 3]$	-3.86
F17	$f_{17}(x) = - \sum_{i=1}^4 c_i \exp \left(- \sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2 \right)$	6	$[0, 1]$	-3.32
F18	$f_{18}(x) = - \sum_{i=1}^5 \left[(X - a_i)(X - a_i)^T + c_i \right]^{-1}$	4	$[0, 10]$	-10.1532
F19	$f_{19}(x) = - \sum_{i=1}^7 \left[(X - a_i)(X - a_i)^T + c_i \right]^{-1}$	4	$[0, 10]$	-10.4028
F20	$f_{20}(x) = - \sum_{i=1}^{10} \left[(X - a_i)(X - a_i)^T + c_i \right]^{-1}$	4	$[0, 10]$	-10.5363

Table A4. Description of CEC-2014 benchmark.

Type	ID	Functions	Global Opt.	Domain
Unimodal function	F21 (CF1)	Rotated High Conditioned Elliptic Function	100	$[-100, 100]$
	F22 (CF2)	Rotated Bent Cigar Function	200	$[-100, 100]$
	F23 (CF3)	Rotated Discus Function	300	$[-100, 100]$
Simple multimodal Test functions	F24 (CF4)	Shifted and Rotated Rosenbrock's Function	400	$[-100, 100]$
	F25 (CF5)	Shifted and Rotated Ackley's Function	500	$[-100, 100]$
	F26 (CF6)	Shifted and Rotated Weierstrass Function	600	$[-100, 100]$
	F27 (CF7)	Shifted and Rotated Griewank's Function	700	$[-100, 100]$
	F28 (CF8)	Shifted Rastrigin's Function	800	$[-100, 100]$
	F29 (CF9)	Shifted and Rotated Rastrigin's Function	900	$[-100, 100]$
	F30 (CF10)	Shifted Schwefel's Function	1000	$[-100, 100]$
	F31 (CF11)	Shifted and Rotated Schwefel's Function	1100	$[-100, 100]$
	F32 (CF12)	Shifted and Rotated Katsuura Function	1200	$[-100, 100]$
	F33 (CF13)	Shifted and Rotated HappyCat Function	1300	$[-100, 100]$
	F34 (CF14)	Shifted and Rotated HGBat Function	1400	$[-100, 100]$
	F35 (CF15)	Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function	1500	$[-100, 100]$
	F36 (CF16)	Shifted and Rotated Expanded Scaffer's F6 Function	1600	$[-100, 100]$
Hybrid test functions	F37 (CF17)	Hybrid Function 1	1700	$[-100, 100]$
	F38 (CF18)	Hybrid Function 2	1800	$[-100, 100]$
	F39 (CF19)	Hybrid Function 3	1900	$[-100, 100]$
	F40 (CF20)	Hybrid Function 4	2000	$[-100, 100]$
	F41 (CF17)	Hybrid Function 5	2100	$[-100, 100]$
	F42 (CF18)	Hybrid Function 6	2200	$[-100, 100]$
Composition test functions	F43 (CF23)	Composition Function 1	2300	$[-100, 100]$
	F44 (CF24)	Composition Function 2	2400	$[-100, 100]$
	F45 (CF25)	Composition Function 3	2500	$[-100, 100]$
	F46 (CF26)	Composition Function 4	2600	$[-100, 100]$
	F47 (CF27)	Composition Function 5	2700	$[-100, 100]$
	F48 (CF28)	Composition Function 6	2800	$[-100, 100]$
	F49 (CF29)	Composition Function 7	2900	$[-100, 100]$
	F50 (CF30)	Composition Function 8	3000	$[-100, 100]$

Table A5. Description of CEC-2017 benchmark.

Type	ID	Functions	Global Opt.	Domain
Unimodal function	F51 (CF1)	Shifted and Rotated Bent Cigar Function	100	[−100,100]
	F52 (CF3)	Shifted and Rotated Zakharov Function	300	[−100,100]
Simple multimodal Test functions	F53 (CF4)	Shifted and Rotated Rosenbrock's Function	400	[−100,100]
	F54 (CF5)	Shifted and Rotated Rastrigin's Function	500	[−100,100]
	F55 (CF6)	Shifted and Rotated Expanded Scaffer's Function	600	[−100,100]
	F56 (CF7)	Shifted and Rotated Lunacek Bi_Rastrigin Function	700	[−100,100]
	F57 (CF8)	Shifted and Rotated Non-Continuous Rastrigin's Function	800	[−100,100]
	F58 (CF9)	Shifted and Rotated Levy Function	900	[−100,100]
	F59 (CF10)	Shifted and Rotated Schwefel's Function	1000	[−100,100]
Hybrid test functions	F60 (CF11)	Hybrid Function 1	1100	[−100,100]
	F61 (CF12)	Hybrid Function 2	1200	[−100,100]
	F62 (CF13)	Hybrid Function 3	1300	[−100,100]
	F63 (CF14)	Hybrid Function 4	1400	[−100,100]
	F64 (CF15)	Hybrid Function 5	1500	[−100,100]
	F65 (CF16)	Hybrid Function 6	1600	[−100,100]
	F66 (CF17)	Hybrid Function 7	1700	[−100,100]
	F67 (CF18)	Hybrid Function 8	1800	[−100,100]
	F68 (CF19)	Hybrid Function 9	1900	[−100,100]
	F69 (CF20)	Hybrid Function 10	2000	[−100,100]
Composition test functions	F70 (CF21)	Composition Function 1	2100	[−100,100]
	F71 (CF22)	Composition Function 2	2200	[−100,100]
	F72 (CF23)	Composition Function 3	2300	[−100,100]
	F73 (CF24)	Composition Function 4	2400	[−100,100]
	F74 (CF25)	Composition Function 5	2500	[−100,100]
	F75 (CF26)	Composition Function 6	2600	[−100,100]
	F76 (CF27)	Composition Function 7	2700	[−100,100]
	F77 (CF28)	Composition Function 8	2800	[−100,100]
	F78 (CF29)	Composition Function 9	2900	[−100,100]
	F79 (CF30)	Composition Function 10	3000	[−100,100]

Table A6. Description of CEC2020 benchmarks.

Type	ID	Functions	Global Opt.	Domain
Unimodal	F80 (CF1)	Shifted and Rotated Bent Cigar Function	100	[−100,100]
multimodal	F81 (CF2)	Shifted and Rotated Lunacek Bi_Rastrigin Function	700	[−100,100]
	F82 (CF3)	Hybrid Function 1	1100	[−100,100]
Hybrid	F83 (CF4)	Hybrid Function 2	1700	[−100,100]
	F84 (CF5)	Hybrid Function 3	1900	[−100,100]
	F85 (CF6)	Hybrid Function 4	2100	[−100,100]
	F86 (CF7)	Hybrid Function 5	1600	[−100,100]
Composition	F87 (CF8)	Composition Function 1	2200	[−100,100]
	F88 (CF9)	Composition Function 2	2400	[−100,100]
	F89 (CF10)	Composition Function 3	2500	[−100,100]

Table A7. Description of CEC2022 benchmark.

Type	No.	Functions	Global Opt.	Domain
Unimodal function	F90	Shifted and full Rotated Zakharov Function	300	[−100,100]
Basic functions	F91	Shifted and full Rotated Rosenbrock's Function	400	[−100,100]
	F92	Shifted and full Rotated Expanded Schaffer's f6 Function	600	[−100,100]
	F93	Shifted and full Rotated Non-continuous Rastrigin's Function	800	[−100,100]
	F94	Shifted and Rotated Levy Function	900	[−100,100]

Table A7. Cont.

Type	No.	Functions	Global Opt.	Domain
Hybrid functions	F95	Hybrid function 1 (N = 3)	1800	[−100,100]
	F96	Hybrid function 2 (N = 6)	2000	[−100,100]
	F97	Hybrid function 3 (N = 5)	2200	[−100,100]
Composite functions	F98	Composite function 1 (N = 5)	2300	[−100,100]
	F99	Composite function 2 (N = 4)	2400	[−100,100]
	F100	Composite function 3 (N = 5)	2600	[−100,100]
	F101	Composite function 4 (N = 6)	2700	[−100,100]

References

1. Fausto, F.; Reyna-Orta, A.; Cuevas, E.; Andrade, Á.G.; Perez-Cisneros, M. From ants to whales: Metaheuristics for all tastes. *Artif. Intell. Rev.* **2020**, *53*, 753–810. [\[CrossRef\]](#)
2. Ross, O.H.M. A review of quantum-inspired metaheuristics: Going from classical computers to real quantum computers. *IEEE Access* **2019**, *8*, 814–838. [\[CrossRef\]](#)
3. Guo, Y.-N.; Zhang, X.; Gong, D.-W.; Zhang, Z.; Yang, J.-J. Novel interactive preference-based multiobjective evolutionary optimization for bolt supporting networks. *IEEE Trans. Evol. Comput.* **2019**, *24*, 750–764. [\[CrossRef\]](#)
4. Guo, Y.; Yang, H.; Chen, M.; Cheng, J.; Gong, D. Ensemble prediction-based dynamic robust multi-objective optimization methods. *Swarm Evol. Comput.* **2019**, *48*, 156–171. [\[CrossRef\]](#)
5. Ji, J.-J.; Guo, Y.-N.; Gao, X.-Z.; Gong, D.-W.; Wang, Y.-P. Q-Learning-Based Hyperheuristic Evolutionary Algorithm for Dynamic Task Allocation of Crowdsensing. *IEEE Trans. Cybern.* **2021**, 34606469. [\[CrossRef\]](#)
6. Loubière, P.; Jourdan, A.; Siarry, P.; Chelouah, R. A sensitivity analysis indicator to adapt the shift length in a metaheuristic. In Proceedings of the 2020 IEEE Congress on Evolutionary Computation (CEC), Glasgow, UK, 19–24 July 2020; pp. 1–6.
7. Gao, K.-Z.; He, Z.M.; Huang, Y.; Duan, P.-Y.; Suganthan, P.N. A survey on meta-heuristics for solving disassembly line balancing, planning and scheduling problems in remanufacturing. *Swarm Evol. Comput.* **2020**, *57*, 100719. [\[CrossRef\]](#)
8. Li, J.; Lei, H.; Alavi, A.H.; Wang, G.-G. Elephant herding optimization: Variants, hybrids, and applications. *Mathematics* **2020**, *8*, 1415. [\[CrossRef\]](#)
9. Feng, Y.; Deb, S.; Wang, G.-G.; Alavi, A.H. Monarch butterfly optimization: A comprehensive review. *Expert Syst. Appl.* **2021**, *168*, 114418. [\[CrossRef\]](#)
10. Li, M.; Wang, G.-G. A Review of Green Shop Scheduling Problem. *Inf. Sci.* **2022**, *589*, 478–496. [\[CrossRef\]](#)
11. Li, W.; Wang, G.-G.; Gandomi, A.H. A survey of learning-based intelligent optimization algorithms. *Arch. Comput. Methods Eng.* **2021**, *28*, 3781–3799. [\[CrossRef\]](#)
12. Han, J.-H.; Choi, D.-J.; Park, S.-U.; Hong, S.-K. Hyperparameter optimization for multi-layer data input using genetic algorithm. In Proceedings of the 2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA), Bangkok, Thailand, 16–21 April 2020; pp. 701–704.
13. Tang, Y.; Jia, H.; Verma, N. Reducing energy of approximate feature extraction in heterogeneous architectures for sensor inference via energy-aware genetic programming. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2020**, *67*, 1576–1587. [\[CrossRef\]](#)
14. Rechenberg, I. Evolutionsstrategien. In *Simulationmethoden in der Medizin und Biologie*; Springer: New York, NY, USA, 1978; pp. 83–114.
15. Dasgupta, D.; Michalewicz, Z. *Evolutionary Algorithms in Engineering Applications*; Springer Science & Business Media: Berlin, Germany, 2013.
16. Opara, K.R.; Arabas, J. Differential Evolution: A survey of theoretical analyses. *Swarm Evol. Comput.* **2019**, *44*, 546–558. [\[CrossRef\]](#)
17. Sharif, M.; Amin, J.; Raza, M.; Yasmin, M.; Satapathy, S.C. An integrated design of particle swarm optimization (PSO) with fusion of features for detection of brain tumor. *Pattern Recognit. Lett.* **2020**, *129*, 150–157. [\[CrossRef\]](#)
18. 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; pp. 1942–1948.
19. Tsipianitis, A.; Tsompanakis, Y. Improved Cuckoo Search algorithmic variants for constrained nonlinear optimization. *Adv. Eng. Softw.* **2020**, *149*, 102865. [\[CrossRef\]](#)
20. Adithiyaa, T.; Chandramohan, D.; Sathish, T. Flower Pollination Algorithm for the Optimization of Stair casting parameter for the preparation of AMC. *Mater. Today Proc.* **2020**, *21*, 882–886. [\[CrossRef\]](#)
21. Faramarzi, A.; Heidarinejad, M.; Mirjalili, S.; Gandomi, A.H. Marine Predators Algorithm: A nature-inspired metaheuristic. *Expert Syst. Appl.* **2020**, *152*, 113377. [\[CrossRef\]](#)
22. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Futur. Gener. Comput. Syst.* **2019**, *97*, 849–872. [\[CrossRef\]](#)
23. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191. [\[CrossRef\]](#)
24. Połap, D.; Woźniak, M. Red fox optimization algorithm. *Expert Syst. Appl.* **2021**, *166*, 114107. [\[CrossRef\]](#)
25. Zhang, M.; Wen, G.; Yang, J. Duck swarm algorithm: A novel swarm intelligence algorithm. *arXiv* **2021**, arXiv:2112.13508.

26. Braik, M.S. Chameleon Swarm Algorithm: A bio-inspired optimizer for solving engineering design problems. *Expert Syst. Appl.* **2021**, *174*, 114685. [\[CrossRef\]](#)
27. Abdollahzadeh, B.; Gharehchopogh, F.S.; Mirjalili, S. Artificial gorilla troops optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems. *Int. J. Intell. Syst.* **2021**, *36*, 5887–5958. [\[CrossRef\]](#)
28. 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; pp. 854–858.
29. Shamsaldin, A.S.; Rashid, T.A.; Al-Rashid Agha, R.A.; Al-Salihi, N.K.; Mohammadi, M. Donkey and smuggler optimization algorithm: A collaborative working approach to path finding. *J. Comput. Des. Eng.* **2019**, *6*, 562–583. [\[CrossRef\]](#)
30. Bolaji, A.L.; Al-Betar, M.A.; Awadallah, M.A.; Khader, A.T.; Abualigah, L. A comprehensive review: Krill Herd algorithm (KH) and its applications. *Appl. Soft Comput.* **2016**, *49*, 437–446. [\[CrossRef\]](#)
31. Wang, G.-G.; Deb, S.; Coelho, L.d.S. Elephant herding optimization. In Proceedings of the 2015 3rd International Symposium on Computational and Business Intelligence (ISCBI), Bali, Indonesia, 7–9 December 2015; pp. 1–5.
32. Yang, C.; Tu, X.; Chen, J. Algorithm of marriage in honey bees optimization based on the wolf pack search. In Proceedings of the 2007 International Conference on Intelligent Pervasive Computing (IPC 2007), Jeju, Korea, 11–13 October 2007; pp. 462–467.
33. Oftadeh, R.; Mahjoob, M.J.; Shariatpanahi, M. A novel meta-heuristic optimization algorithm inspired by group hunting of animals: Hunting search. *Comput. Math. Appl.* **2010**, *60*, 2087–2098. [\[CrossRef\]](#)
34. Mucherino, A.; Seref, O. Monkey search: A novel metaheuristic search for global optimization. In Proceedings of the Conference on Data Mining, System Analysis and Optimization in Biomedicine, Gainesville, FL, USA, 28–30 March 2007; pp. 162–173.
35. 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; pp. 86–94.
36. MiarNaeimi, F.; Azizyan, G.; Rashki, M. Horse herd optimization algorithm: A nature-inspired algorithm for high-dimensional optimization problems. *Knowl.-Based Syst.* **2021**, *213*, 106711. [\[CrossRef\]](#)
37. Wang, G.-G. Moth search algorithm: A bio-inspired metaheuristic algorithm for global optimization problems. *Memetic Comput.* **2018**, *10*, 151–164. [\[CrossRef\]](#)
38. Wang, G.G.; Deb, S.; Coelho, L.D.S. Earthworm optimisation algorithm: A bio-inspired metaheuristic algorithm for global optimisation problems. *Int. J. Bio-Inspired Comput.* **2018**, *12*, 1–22. [\[CrossRef\]](#)
39. Wang, G.-G.; Deb, S.; Cui, Z. Monarch butterfly optimization. *Neural Comput. Appl.* **2019**, *31*, 1995–2014. [\[CrossRef\]](#)
40. Li, S.; Chen, H.; Wang, M.; Heidari, A.A.; Mirjalili, S. Slime mould algorithm: A new method for stochastic optimization. *Future Gener. Comput. Syst.* **2020**, *111*, 300–323. [\[CrossRef\]](#)
41. Zhang, J.; Wang, J.S. Improved whale optimization algorithm based on nonlinear adaptive weight and golden sine operator. *IEEE Access* **2020**, *8*, 77013–77048. [\[CrossRef\]](#)
42. Houssein, E.H.; Saad, M.R.; Hashim, F.A.; Shaban, H.; Hassaballah, M. Lévy flight distribution: A new metaheuristic algorithm for solving engineering optimization problems. *Eng. Appl. Artif. Intell.* **2020**, *94*, 103731. [\[CrossRef\]](#)
43. Geem, Z.W.; Kim, J.H.; Loganathan, G.V. A new heuristic optimization algorithm: Harmony search. *Simulation* **2001**, *76*, 60–68. [\[CrossRef\]](#)
44. Naik, A.; Satapathy, S.C. Past present future: A new human-based algorithm for stochastic optimization. *Soft Comput.* **2021**, *25*, 12915–12976. [\[CrossRef\]](#)
45. Askari, Q.; Younas, I.; Saeed, M. Political Optimizer: A novel socio-inspired meta-heuristic for global optimization. *Knowl.-Based Syst.* **2020**, *195*, 105709. [\[CrossRef\]](#)
46. Shi, Y. Brain storm optimization algorithm. In Proceedings of the International Conference in Swarm Intelligence, Chongqing, China, 12–15 June 2011; pp. 303–309.
47. Ghorbani, N.; Babaei, E. Exchange market algorithm. *Appl. Soft Comput.* **2014**, *19*, 177–187. [\[CrossRef\]](#)
48. Kashan, A.H. League Championship Algorithm (LCA): An algorithm for global optimization inspired by sport championships. *Appl. Soft Comput.* **2014**, *16*, 171–200. [\[CrossRef\]](#)
49. Moosavi, S.H.S.; Bardsiri, V.K. Poor and rich optimization algorithm: A new human-based and multi populations algorithm. *Eng. Appl. Artif. Intell.* **2019**, *86*, 165–181. [\[CrossRef\]](#)
50. Dehghani, M.; Trojovská, E.; Trojovský, P. Driving Training-Based Optimization: A New Human-Based Metaheuristic Algorithm for Solving Optimization Problems. *Res. Square* **2022**. [\[CrossRef\]](#)
51. Mohamed, A.W.; Hadi, A.A.; Mohamed, A. Gaining-sharing knowledge based algorithm for solving optimization problems: A novel nature-inspired algorithm. *Int. J. Mach. Learn. Cybern.* **2020**, *11*, 1501–1529.
52. Atashpaz-Gargari, E.; Lucas, C. Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. In Proceedings of the 2007 IEEE Congress on Evolutionary Computation, Singapore, 25–28 September 2007; pp. 4661–4667.
53. Moosavian, N.; Roodsari, B.K. Soccer league competition algorithm: A novel meta-heuristic algorithm for optimal design of water distribution networks. *Swarm Evol. Comput.* **2014**, *17*, 14–24. [\[CrossRef\]](#)
54. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M. Optimization by simulated annealing. *Science* **1983**, *220*, 671–680. [\[CrossRef\]](#) [\[PubMed\]](#)
55. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. GSA: A gravitational search algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [\[CrossRef\]](#)
56. Kaveh, A.; Talatahari, S. A novel heuristic optimization method: Charged system search. *Acta Mech.* **2010**, *213*, 267–289. [\[CrossRef\]](#)
57. Erol, O.K.; Eksin, I. A new optimization method: Big bang–big crunch. *Adv. Eng. Softw.* **2006**, *37*, 106–111. [\[CrossRef\]](#)
58. Xie, L.; Zeng, J.; Cui, Z. General framework of artificial physics optimization algorithm. In Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), Coimbatore, India, 9–11 December 2009; pp. 1321–1326.

59. Shah-Hosseini, H. Principal components analysis by the galaxy-based search algorithm: A novel metaheuristic for continuous optimisation. *Int. J. Comput. Sci. Eng.* **2011**, *6*, 132–140.
60. Hatamlou, A. Black hole: A new heuristic optimization approach for data clustering. *Inf. Sci.* **2013**, *222*, 175–184. [\[CrossRef\]](#)
61. Rabanal, P.; Rodríguez, I.; Rubio, F. Using river formation dynamics to design heuristic algorithms. In Proceedings of the International Conference on Unconventional Computation, Kingston, UC, Canada, 13–17 August 2007; pp. 163–177.
62. Hashim, F.A.; Houssein, E.H.; Mabrouk, M.S.; Al-Atabany, W.; Mirjalili, S. Henry gas solubility optimization: A novel physics-based algorithm. *Future Gener. Comput. Syst.* **2019**, *101*, 646–667. [\[CrossRef\]](#)
63. Moghaddam, F.F.; Moghaddam, R.F.; Cheriet, M. Curved space optimization: A random search based on general relativity theory. *arXiv* **2012**, arXiv:1208.2214.
64. Formato, R. Central force optimization. *Prog. Electromagn. Res.* **2007**, *77*, 425–491. [\[CrossRef\]](#)
65. Eskandar, H.; Sadollah, A.; Bahreininejad, A.; Hamdi, M. Water cycle algorithm—A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Comput. Struct.* **2012**, *110*, 151–166. [\[CrossRef\]](#)
66. Zheng, Y.-J. Water wave optimization: A new nature-inspired metaheuristic. *Comput. Oper. Res.* **2015**, *55*, 1–11. [\[CrossRef\]](#)
67. Kaveh, A.; Khayatazad, M. A new meta-heuristic method: Ray optimization. *Comput. Struct.* **2012**, *112–113*, 283–294. [\[CrossRef\]](#)
68. Webster, B.; Bernhard, P.J. *A Local Search Optimization Algorithm Based on Natural Principles of Gravitation*; Florida Institute of Technology: Melbourne, FL, USA, 2003.
69. Du, H.; Wu, X.; Zhuang, J. Small-world optimization algorithm for function optimization. In Proceedings of the International Conference on Natural Computation, Xi'an, China, 24–28 September 2006; pp. 264–273.
70. Mirjalili, S.; Mirjalili, S.M.; Hatamlou, A. Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Appl.* **2016**, *27*, 495–513. [\[CrossRef\]](#)
71. Shah-Hosseini, H. The intelligent water drops algorithm: A nature-inspired swarm-based optimization algorithm. *Int. J. Bio-Inspired Comput.* **2009**, *1*, 71–79. [\[CrossRef\]](#)
72. Chuang, C.-L.; Jiang, J.-A. Integrated radiation optimization: Inspired by the gravitational radiation in the curvature of space-time. In Proceedings of the 2007 IEEE Congress on Evolutionary Computation, Singapore, 25–28 September 2007; pp. 3157–3164.
73. Hsiao, Y.-T.; Chuang, C.-L.; Jiang, J.-A.; Chien, C.-C. A novel optimization algorithm: Space gravitational optimization. In Proceedings of the 2005 IEEE international Conference on Systems, Man and Cybernetics, Waikoloa, HI, USA, 10–12 October 2005; pp. 2323–2328.
74. Javidy, B.; Hatamlou, A.; Mirjalili, S. Ions motion algorithm for solving optimization problems. *Appl. Soft Comput.* **2015**, *32*, 72–79. [\[CrossRef\]](#)
75. Birbil, Ş.İ.; Fang, S.-C. An electromagnetism-like mechanism for global optimization. *J. Glob. Optim.* **2003**, *25*, 263–282. [\[CrossRef\]](#)
76. Faramarzi, A.; Heidarinejad, M.; Stephens, B.; Mirjalili, S. Equilibrium optimizer: A novel optimization algorithm. *Knowl.-Based Syst.* **2020**, *191*, 105190. [\[CrossRef\]](#)
77. Shen, J.; Li, J. The principle analysis of light ray optimization algorithm. In Proceedings of the 2010 Second International Conference on Computational Intelligence and Natural Computing, Washington, DC, USA, 23–24 October 2010; pp. 154–157.
78. Hashim, F.A.; Hussain, K.; Houssein, E.H.; Mabrouk, M.S.; Al-Atabany, W. Archimedes optimization algorithm: A new metaheuristic algorithm for solving optimization problems. *Appl. Intell.* **2021**, *51*, 1531–1551. [\[CrossRef\]](#)
79. Abdechiri, M.; Meybodi, M.R.; Bahrami, H. Gases Brownian motion optimization: An algorithm for optimization (GBMO). *Appl. Soft Comput.* **2013**, *13*, 2932–2946. [\[CrossRef\]](#)
80. Alatas, B. ACROA: Artificial chemical reaction optimization algorithm for global optimization. *Expert Syst. Appl.* **2011**, *38*, 13170–13180. [\[CrossRef\]](#)
81. Siddique, N.; Adeli, H. *Nature-Inspired Computing: Physics-and Chemistry-Based Algorithms*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2017.
82. Tanyildizi, E.; Demir, G. Golden sine algorithm: A novel math-inspired algorithm. *Adv. Electr. Comput. Eng.* **2017**, *17*, 71–78. [\[CrossRef\]](#)
83. Salem, S.A. BOA: A novel optimization algorithm. In Proceedings of the 2012 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 21–23 August 2017; pp. 1–5.
84. Mirjalili, S. SCA: A sine cosine algorithm for solving optimization problems. *Knowl.-Based Syst.* **2016**, *96*, 120–133. [\[CrossRef\]](#)
85. Mara, S.T.W.; Norcahyo, R.; Jodiawan, P.; Lusiantoro, L.; Rifai, A.P. A survey of adaptive large neighborhood search algorithms and applications. *Comput. Oper. Res.* **2022**, *146*, 105903. [\[CrossRef\]](#)
86. Pisinger, D.; Ropke, S. Large neighborhood search. In *Handbook of Metaheuristics*; Springer: New York, NY, USA, 2010; pp. 399–419.
87. Ahuja, R.K.; Ergun, Ö.; Orlin, J.B.; Punnen, A.P. A survey of very large-scale neighborhood search techniques. *Discret. Appl. Math.* **2002**, *123*, 75–102. [\[CrossRef\]](#)
88. Feo, T.A.; Resende, M. Greedy randomized adaptive search procedures. *J. Glob. Optim.* **1995**, *6*, 109–133. [\[CrossRef\]](#)
89. Lee, D.-H.; Chen, J.H.; Cao, J.X.; Review, T. The continuous berth allocation problem: A greedy randomized adaptive search solution. *Transp. Res. Part E Logist. Transp. Rev.* **2010**, *46*, 1017–1029. [\[CrossRef\]](#)
90. Zamani, H.; Nadimi-Shahraki, M.H.; Gandomi, A.H. Starling murmuration optimizer: A novel bio-inspired algorithm for global and engineering optimization. *Comput. Methods Appl. Mech. Eng.* **2022**, *392*, 114616. [\[CrossRef\]](#)
91. Hashim, F.A.; Hussien, A.G. Snake Optimizer: A novel meta-heuristic optimization algorithm. *Knowl.-Based Syst.* **2022**, *242*, 108320. [\[CrossRef\]](#)

92. Abualigah, L.; Abd Elaziz, M.; Sumari, P.; Geem, Z.W.; Gandomi, A.H. Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer. *Expert Syst. Appl.* **2022**, *191*, 116158. [\[CrossRef\]](#)
93. Zitouni, F.; Harous, S.; Belkeram, A.; Hammou, L.E.B. The archerfish hunting optimizer: A novel metaheuristic algorithm for global optimization. *Arab. J. Sci. Eng.* **2022**, *47*, 2513–2553. [\[CrossRef\]](#)
94. Daliri, A.; Asghari, A.; Azgomi, H.; Alimoradi, M. The water optimization algorithm: A novel metaheuristic for solving optimization problems. *Appl. Intell.* **2022**, 1–40. [\[CrossRef\]](#)
95. Oyelade, O.N.; Ezugwu, A.E.-S.; Mohamed, T.I.A.; Abualigah, L. Ebola optimization search algorithm: A new nature-inspired metaheuristic optimization algorithm. *IEEE Access* **2022**, *10*, 16150–16177. [\[CrossRef\]](#)
96. Zhong, C.; Li, G.; Meng, Z. Beluga whale optimization: A novel nature-inspired metaheuristic algorithm. *Knowl.-Based Syst.* **2022**, *251*, 109215. [\[CrossRef\]](#)
97. Eslami, N.; Yazdani, S.; Mirzaei, M.; Hadavandi, E. Aphid-Ant Mutualism: A novel nature-inspired metaheuristic algorithm for solving optimization problems. *Math. Comput. Simul.* **2022**, *201*, 362–395. [\[CrossRef\]](#)
98. Qais, M.H.; Hasanien, H.M.; Turkey, R.A.; Alghuwainem, S.; Tostado-Véliz, M.; Jurado, F. Circle Search Algorithm: A Geometry-Based Metaheuristic Optimization Algorithm. *Mathematics* **2022**, *10*, 1626. [\[CrossRef\]](#)
99. Trojovský, P.; Dehghani, M. Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications. *Sensors* **2022**, *22*, 855. [\[CrossRef\]](#) [\[PubMed\]](#)
100. Kivi, M.E.; Majidnezhad, V.J.; Computing, H. A novel swarm intelligence algorithm inspired by the grazing of sheep. *J. Ambient. Intell. Humaniz. Comput.* **2022**, *13*, 1201–1213. [\[CrossRef\]](#)
101. Pan, J.-S.; Zhang, L.-G.; Wang, R.-B.; Snášel, V.; Chu, S.-C. Gannet optimization algorithm: A new metaheuristic algorithm for solving engineering optimization problems. *Math. Comput. Simul.* **2022**, *202*, 343–373. [\[CrossRef\]](#)
102. Ezugwu, A.E.; Agushaka, J.O.; Abualigah, L.; Mirjalili, S.; Gandomi, A.H. Prairie dog optimization algorithm. *Neural Comput. Appl.* **2022**, 1–49. [\[CrossRef\]](#)
103. Emami, H. Stock exchange trading optimization algorithm: A human-inspired method for global optimization. *J. Supercomput.* **2022**, *78*, 2125–2174. [\[CrossRef\]](#) [\[PubMed\]](#)
104. 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. [\[CrossRef\]](#)
105. Askari, Q.; Saeed, M.; Younas, I. Heap-based optimizer inspired by corporate rank hierarchy for global optimization. *Expert Syst. Appl.* **2020**, *161*, 113702. [\[CrossRef\]](#)
106. Abdollahzadeh, B.; Gharehchopogh, F.S.; Mirjalili, S. African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems. *Comput. Ind. Eng.* **2021**, *158*, 107408. [\[CrossRef\]](#)
107. Zamani, H.; Nadimi-Shahraki, M.H.; Gandomi, A.H. QANA: Quantum-based avian navigation optimizer algorithm. *Eng. Appl. Artif. Intell.* **2021**, *104*, 104314. [\[CrossRef\]](#)
108. Tu, J.; Chen, H.; Wang, M.; Gandomi, A.H. The colony predation algorithm. *J. Bionic Eng.* **2021**, *18*, 674–710. [\[CrossRef\]](#)
109. Davies, O.; Wannell, J.; Inglesfield, J. *The Rainbow*; Cambridge University Press: Cambridge, UK, 2006; Volume 37, pp. 17–21.
110. Adam, J.A. The mathematical physics of rainbows and glories. *Phys. Rep.* **2002**, *356*, 229–365. [\[CrossRef\]](#)
111. Suzuki, M.; Suzuki, I.S. Physics of rainbow. *Phys. Teach.* **2010**, *12*, 283–286.
112. Buchwald, J.Z. Descartes's experimental journey past the prism and through the invisible world to the rainbow. *Ann. Sci.* **2008**, *65*, 1–46. [\[CrossRef\]](#)
113. Zhou, J.; Fang, Y.; Wang, J.; Zhu, L.; Wriedt, T. Rainbow pattern analysis of a multilayered sphere for optical diagnostic of a heating droplet. *Opt. Commun.* **2019**, *441*, 113–120. [\[CrossRef\]](#)
114. Wu, Y.; Li, H.; Wu, X.; Gréhan, G.; Mädler, L.; Crua, C. Change of evaporation rate of single monocomponent droplet with temperature using time-resolved phase rainbow refractometry. *Proc. Combust. Inst.* **2019**, *37*, 3211–3218. [\[CrossRef\]](#)
115. Wu, Y.; Li, C.; Crua, C.; Wu, X.; Saengkaew, S.; Chen, L.; Gréhan, G.; Cen, K. Primary rainbow of high refractive index particle ($1.547 < n < 2$) has refraction ripples. *Opt. Commun.* **2018**, *426*, 237–241.
116. Yu, H.; Xu, F.; Tropea, C. Optical caustics associated with the primary rainbow of oblate droplets: Simulation and application in non-sphericity measurement. *Opt. Express* **2013**, *21*, 25761–25771. [\[CrossRef\]](#) [\[PubMed\]](#)
117. Greengard, P.; Rokhlin, V. An algorithm for the evaluation of the incomplete gamma function. *Adv. Comput. Math.* **2019**, *45*, 23–49. [\[CrossRef\]](#)
118. Yang, X.-S. Firefly algorithm, stochastic test functions and design optimisation. *Int. J. Bio-Inspired Comput.* **2010**, *2*, 78–84. [\[CrossRef\]](#)
119. Digalakis, J.G.; Margaritis, K.G. On benchmarking functions for genetic algorithms. *Int. J. Comput. Math.* **2001**, *77*, 481–506. [\[CrossRef\]](#)
120. Molga, M.; Smutnicki, C. Test functions for optimization needs. *Test Funct. Optim. Needs* **2005**, *101*, 48.
121. De Barros, R.S.M.; Hidalgo, J.I.G.; de Lima Cabral, D.R. Wilcoxon rank sum test drift detector. *Neurocomputing* **2018**, *275*, 1954–1963. [\[CrossRef\]](#)
122. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [\[CrossRef\]](#)
123. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [\[CrossRef\]](#)
124. Ahmadianfar, I.; Bozorg-Haddad, O.; Chu, X. Gradient-based optimizer: A new metaheuristic optimization algorithm. *Inf. Sci.* **2020**, *540*, 131–159. [\[CrossRef\]](#)
125. Ahmadianfar, I.; Heidari, A.A.; Gandomi, A.H.; Chu, X.; Chen, H. RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method. *Expert Syst. Appl.* **2021**, *181*, 115079. [\[CrossRef\]](#)

126. Awad, N.H.; Ali, M.Z.; Suganthan, P.N. Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems. In Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC), San Sebastián, Spain, 5–8 June 2017; pp. 372–379.
127. Nadimi-Shahraki, M.H.; Zamani, H. DMDE: Diversity-maintained multi-trial vector differential evolution algorithm for non-decomposition large-scale global optimization. *Expert Syst. Appl.* **2022**, *198*, 116895. [\[CrossRef\]](#)
128. Seck-Tuoh-Mora, J.C.; Hernandez-Romero, N.; Lagos-Eulogio, P.; Medina-Marin, J.; Zuñiga-Peña, N.S. A continuous-state cellular automata algorithm for global optimization. *Expert Syst. Appl.* **2021**, *177*, 114930. [\[CrossRef\]](#)
129. Coello, C.A.C. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art. *Comput. Methods Appl. Mech. Eng.* **2002**, *191*, 1245–1287. [\[CrossRef\]](#)
130. Rao, R.V.; Waghmare, G.G. A new optimization algorithm for solving complex constrained design optimization problems. *Comput. Methods Appl. Mech. Eng.* **2017**, *49*, 60–83. [\[CrossRef\]](#)
131. Mezura-Montes, E.; Coello, C.A.C. An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. *Int. J. Gen. Syst.* **2008**, *37*, 443–473. [\[CrossRef\]](#)
132. Ali, M.Z.; Reynolds, R.G. Cultural algorithms: A Tabu search approach for the optimization of engineering design problems. *Soft Comput.* **2014**, *18*, 1631–1644. [\[CrossRef\]](#)
133. Ray, T.; Saini, P. Engineering design optimization using a swarm with an intelligent information sharing among individuals. *Eng. Optim.* **2001**, *33*, 735–748. [\[CrossRef\]](#)
134. Parsopoulos, K.E.; Vrahatis, M.N. Unified particle swarm optimization for solving constrained engineering optimization problems. In *International Conference on Natural Computation*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 582–591.
135. Kim, M.-S.; Kim, J.-R.; Jeon, J.-Y.; Choi, D.-H. Efficient mechanical system optimization using two-point diagonal quadratic approximation in the nonlinear intervening variable space. *KSME Int. J.* **2001**, *15*, 1257–1267. [\[CrossRef\]](#)
136. Kaveh, A.; Talatahari, S. Hybrid charged system search and particle swarm optimization for engineering design problems. *Eng. Comput.* **2011**, *28*, 423–440. [\[CrossRef\]](#)
137. Mani, A.; Patvardhan, C. An adaptive quantum evolutionary algorithm for engineering optimization problems. *Int. J. Comput. Appl.* **2010**, *1*, 43–48. [\[CrossRef\]](#)
138. Lisboa, R.; Yasojima, E.K.; de Oliveira, R.M.S.; Mollinetti, M.A.F.; Teixeira, O.N.; de Oliveira, R.C.L. Parallel genetic algorithm with social interaction for solving constrained global optimization problems. In Proceedings of the 2015 7th International Conference of Soft Computing and Pattern Recognition (SoCPaR), Fukuoka, Japan, 13–15 November 2015; pp. 351–356.
139. Kulkarni, A.J.; Tai, K. Solving constrained optimization problems using probability collectives and a penalty function approach. *Int. J. Comput. Intell. Appl.* **2011**, *10*, 445–470. [\[CrossRef\]](#)
140. Dimopoulos, G.G. Mixed-variable engineering optimization based on evolutionary and social metaphors. *Comput. Methods Appl. Mech. Eng.* **2007**, *196*, 803–817. [\[CrossRef\]](#)
141. Sedhom, B.E.; El-Saadawi, M.M.; Hatata, A.Y.; Alsayyari, A.S.; Systems, E. Hierarchical control technique-based harmony search optimization algorithm versus model predictive control for autonomous smart microgrids. *Int. J. Electr. Power Energy Syst.* **2020**, *115*, 105511. [\[CrossRef\]](#)
142. Gandomi, A.H.; Yang, X.-S.; Alavi, A.H. Mixed variable structural optimization using firefly algorithm. *Comput. Struct.* **2011**, *89*, 2325–2336. [\[CrossRef\]](#)
143. Zhang, M.; Luo, W.; Wang, X. Differential evolution with dynamic stochastic selection for constrained optimization. *Inf. Sci.* **2008**, *178*, 3043–3074. [\[CrossRef\]](#)
144. He, Q.; Wang, L. A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. *Appl. Math. Comput.* **2007**, *186*, 1407–1422. [\[CrossRef\]](#)
145. Dos Santos Coelho, L. Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. *Expert Syst. Appl.* **2010**, *37*, 1676–1683. [\[CrossRef\]](#)
146. Kaveh, A.; Bakhshpoori, T. Water evaporation optimization: A novel physically inspired optimization algorithm. *Comput. Struct.* **2016**, *167*, 69–85. [\[CrossRef\]](#)
147. Gandomi, A.H.; Yang, X.-S.; Alavi, A.H.; Talatahari, S. Bat algorithm for constrained optimization tasks. *Neural Comput. Appl.* **2013**, *22*, 1239–1255. [\[CrossRef\]](#)
148. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl.-Based Syst.* **2015**, *89*, 228–249. [\[CrossRef\]](#)
149. Mezura-Montes, E.; Coello, C.A. A simple multimembered evolution strategy to solve constrained optimization problems. *IEEE Trans. Evol. Comput.* **2005**, *9*, 1–17. [\[CrossRef\]](#)
150. Montemurro, M.; Vincenti, A.; Vannucci, P. The automatic dynamic penalisation method (ADP) for handling constraints with genetic algorithms. *Comput. Methods Appl. Mech. Eng.* **2013**, *256*, 70–87. [\[CrossRef\]](#)
151. Mezura-Montes, E.; Coello Coello, C.A.; Velázquez-Reyes, J.; Muñoz-Dávila, L. Multiple trial vectors in differential evolution for engineering design. *Eng. Optim.* **2007**, *39*, 567–589. [\[CrossRef\]](#)
152. Wang, L.; Li, L.-P. An effective differential evolution with level comparison for constrained engineering design. *Struct. Multidiscip. Optim.* **2010**, *41*, 947–963. [\[CrossRef\]](#)
153. Coello, C.A.C.; Montes, E.M. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Adv. Eng. Inform.* **2002**, *16*, 193–203. [\[CrossRef\]](#)