applied sciences

MDPI

Article

Lemurs Optimizer: A New Metaheuristic Algorithm for Global Optimization

Ammar Kamal Abasi [1,*], Sharif Naser Makhadmeh [2], Mohammed Azmi Al-Betar [2], Osama Ahmad Alomari [3], Mohammed A. Awadallah [4,5], Zaid Abdi Alkareem Alyasseri [6,7,8], Iyad Abu Doush [9,10], Ashraf Elnagar [11], Eman H. Alkhammash [12] and Myriam Hadjouni [13]

1 Machine Learning Department, Mohamed bin Zayed University of Artificial Intelligence (MBZUAI), Abu Dhabi P.O. Box 54115, United Arab Emirates
2 Artificial Intelligence Research Center (AIRC), College of Engineering and Information Technology, Ajman University, Ajman P.O. Box 346, United Arab Emirates
3 MLALP Research Group, University of Sharjah, Sharjah P.O. Box 27272, United Arab Emirates
4 Department of Computer Science, Al-Aqsa University, Gaza P.O. Box 4051, Palestine
5 Artificial Intelligence Research Center (AIRC), Ajman University, Ajman P.O. Box 346, United Arab Emirates
6 ECE Department, Faculty of Engineering, University of Kufa, Najaf P.O. Box 21, Iraq
7 Information Technology Research and Development Center (ITRDC), University of Kufa, Najaf 54003, Iraq
8 Department of Business Administration, College of Administrative and Financial Sciences, Imam Ja'afar Al-Sadiq University, Baghdad P.O. Box 9102, Iraq
9 Computer Science Department, Yarmouk University, Irbid P.O. Box 566, Jordan
10 Department of Computing, College of Engineering and Applied Sciences, American University of Kuwait, Salmiya P.O. Box 3323, Kuwait
11 Department of Computer Science, University of Sharjah, Sharjah P.O. Box 27272, United Arab Emirates
12 Department of Computer Science, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia
13 Department of Computer Sciences, College of Computer and Information Science, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
* Correspondence: ammar.abasi@mbzuai.ac.ae; Tel.: +971-50947-3679

Abstract: The Lemur Optimizer (LO) is a novel nature-inspired algorithm we propose in this paper. This algorithm's primary inspirations are based on two pillars of lemur behavior: leap up and dance hub. These two principles are mathematically modeled in the optimization context to handle local search, exploitation, and exploration search concepts. The LO is first benchmarked on twenty-three standard optimization functions. Additionally, the LO is used to solve three real-world problems to evaluate its performance and effectiveness. In this direction, LO is compared to six well-known algorithms: Salp Swarm Algorithm (SSA), Artificial Bee Colony (ABC), Sine Cosine Algorithm (SCA), Bat Algorithm (BA), Flower Pollination Algorithm (FPA), and JAYA algorithm. The findings show that the proposed algorithm outperforms these algorithms in fourteen standard optimization functions and proves the LO's robust performance in managing its exploration and exploitation capabilities, which significantly leads LO towards the global optimum. The real-world experimental findings demonstrate how LO may tackle such challenges competitively.

Keywords: swarm intelligence; metaheuristic; optimization; stochastic optimization; benchmark; LO

Citation: Abasi, A.K.; Makhadmeh, S.N.; Al-Betar, M.A.; Alomari, O.A.; Awadallah, M.A.; Alyasseri, Z.A.A.; Doush, I.A.; Elnagar, A.; Alkhammash, E.H.; Hadjouni, M. Lemurs Optimizer: A New Metaheuristic Algorithm for Global Optimization. *Appl. Sci.* **2022**, *12*, 10057. https://doi.org/10.3390/app121910057

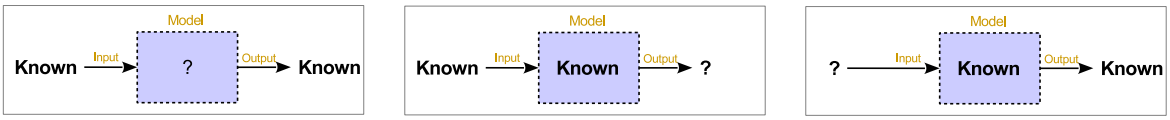
Academic Editor: Giancarlo Mauri

Received: 2 September 2022
Accepted: 29 September 2022
Published: 6 October 2022

1. Introduction

In real life, the optimization problem [1] can be normally classified as a "black box" model which consists of three main components: input, model, and output as shown in Figure 1. In case any component is unknown, a new problem type arises. When the input and output are known and the model is unknown, this type of problem is called a modeling problem where the solution is to find the function that maps the input to the output. This type of problem can be heavily seen in data mining and machine learning domain, especially in prediction and classification problems.

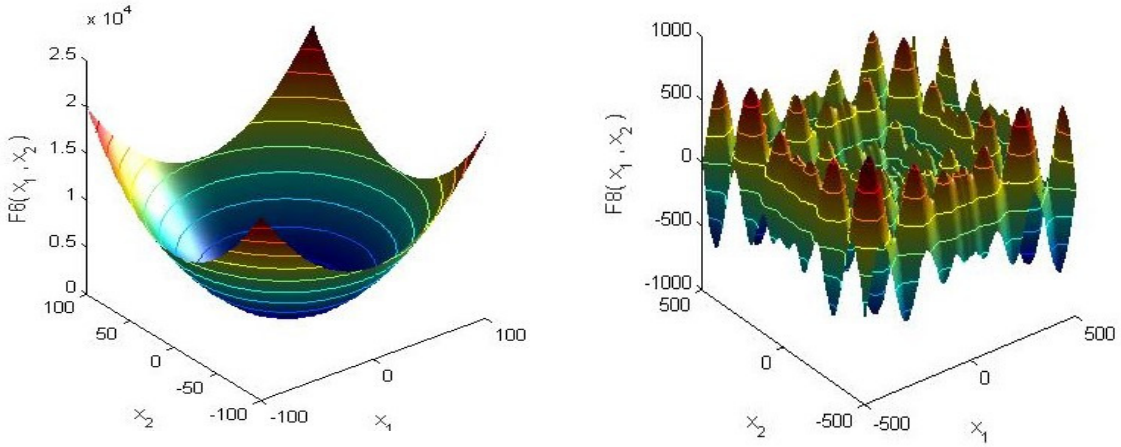
When some inputs and models are known, and the target is to enter these input conditions into the model to determine the output, this problem is known as a simulation problem that can be used in engineering design problems, especially for forecasting. Finally, when the model and the desired output are known and the target is to find the input, this problem is known as the optimization problem. An example of optimization problems is the creating an optimal image quality evaluator [2], feature selection [3], and scheduling [4,5], additive manufacturing [6], renewable energy system [7], etc.



(**a**) modeling (**b**) simulation (**c**) optimization

Figure 1. "black box" models for real-life problems.

In general, optimization problems normally include a set of decision variables as an input to the objective function as a model where the desired output is known or can be measured. The main target is to find the optimal values for the decision variables that results in the minimal or maximum value of the objective function [8]. Based on their value ranges, the alternative combination of decision variables forms a huge search space. The definition of the search space depends solely on the problem characteristics. The problems have either unimodal modal or multimodal search space as can be shown in Figure 2. The complexity of the problems is normally justified based on the search space ruggedness and solution dimensions. The ruggedness of the search space can be related to the problem constraints while solution dimensions can be related to the problem size.



(**a**) Unimodal (Step problem) (**b**) Multimodal (Generalized Schwefel's problem)

Figure 2. 3D search space for optimization functions.

The constraints of the optimization problem determine the movement through the search space. The earliest methods are based on mathematical theories such as integer/linear programming, Simplex Method, etc. The main advantage of these methods for the optimization problem is that they can find the exact solution by conducting an exhaustive search; however, they are normally unworkable with the optimization problem of NP-hard and NP-Complete classes since the polynomial-time required is almost uncountable. As a result, heuristic-based approaches emerged to find an approximate solution for the optimization problem in a reasonable time. The heuristic-based techniques are normally problem-dependent where the problem-specific knowledge is embedded. Examples of these heuristic-based approaches in traveling salesman problems such as 2-opt and 3-opt moves. These heuristics indeed have efficiency in solving the optimization problems in

a small amount of time, although they work as a constraint satisfaction method with less concern for the solution quality.

A new metaphor in evolutionary computation is produced to deal with optimization problems. A population of individuals who competes in an environment with limited resources to survive is the imitation of evolutionary algorithms (EAs). In EAs, the fitter individuals have a better chance to inherit their strong attributes to the next generations. This is known as the Darwinian natural selection of survival-of-the-fittest principle. EAs are general optimization templates that can be applied to a variety of optimization problems through efficient operators for sharing knowledge such as recombination, mutation, and selection controlled by carefully initialized parameters until a "good-enough" individual is obtained. The first generation EAs are Genetic Algorithm (GA) [9], Genetic Programming [10], Evolution Strategy [11], and Biogeography-Based Optimizer [12]. These methods are generated by taking into consideration the type of the problem (i.e., binary, discrete, continuous, perpetration, or structured). In the second generation of EAs, only one optimization template is available for almost all optimization problems. These includes Differential Evolution (DE) [13], Particle Swarm Optimization (PSO) [14].

Nowadays, a plethora of EAs is mostly inspired by natural phenomena related to human behavior, physical wisdom, animal swarm survival strategies, and chemical principles. These methods will be surveyed in the literature review section. The common features of these methods can be summarized as follows: (i) they are population-based, (ii) they have an iterative improvement process, (iii) They turned toward the optimal solution through the special operator(s), (iv) they can explore several search space niches and exploit each niche, (v) they embed the problem-specific knowledge mapping the phenotype to genotype, and (vi) they can provide a suitable balance between diversification and intensification of the problem search space. The main difference between them is their movement way through the search space via their operators where their definitions of the search space niches are varied.

Due to the complex nature of the optimization problem, where there is up-to-now a superior EA can ultimately tackle all optimization problems and excel over all other EAs. This is stated in a pioneer theorem in optimization named No Free Lunch (NFL) where there is no single algorithm that performs better than others in every case or even for the same problem in different instances [15]. Therefore, NFL opens the door for new innovation theories to stem other natural phenomena and propose other intelligent EAs with the ability to efficiently tackle optimization problems with rugged and huge search space.

Although there are many studies in the optimization field, there are still various behaviors in nature that have not been studied yet [16,17]. One of these behaviors is the behavior of lemurs animal in the movement to search for food or escape from other predators. Lemurs can only be found in Madagascar and the neighboring Comoro Islands, which are located off the coast of Mozambique, Africa.

In this paper, the Lemurs Optimizer (LO) is proposed as an evolutionary algorithm. LO is stemmed by its behavior in locomotor behavior: leap up and dance-hup. The LO behavior is formalized in terms of optimization context. LO is evaluated using twenty-three standard optimization functions circulated well in the literature. In addition, some real engineering problems are also used. Initially, the effect of some parameters in LO is studied to decide which is the best configuration. Thereafter, the comparative evaluation is conducted against six well-established methods. The results prove the superiority of LO over other comparative algorithms. For statistical evaluation, the Wilcoxon Mann-Whitney test shows the significance of LO results. In conclusion, LO is a new optimization algorithm that can be applied to a large variety of global optimization problems efficiently.

This paper's remaining subsections are organized as follows: The literature review of the previous natural inspired evolutionary algorithms is summarized in Section 2. The inspiration and procedural steps of the LO algorithm are proposed and described in Section 3. The evaluation of the proposed LO is conducted and the experimental results

are analyzed and compared in Section 4. Finally, in Section 5, the conclusion is presented, as well as various scenarios for future development.

2. Literature Review

Evolutionary Algorithms (EA) are random search algorithms that are inspired by the concept of natural evolution, where this inspiration concept re-formulates into a set of optimization operators that combine to form an optimization algorithm. The population of EA algorithms is a random set of solutions that are used as an initial point for the algorithm, and after the growth of the generations, the genes of the parent individuals are subjected to change or alternation in the process of producing new offspring individuals by recombination and mutation processes, where the natural selection method utilizes the survival-of-the-fittest principle to select these offspring. EA's first natural evolution-inspired algorithm is called GA, which is introduced in 1960 by John Henry Holland [18].

Swarm-based algorithms imitated the swarm collaboration behavior of animals. Particle Swarm Optimization (PSO) [14] is the most popular swarm-based algorithm, which imitates the social behavior of birds. The optimization framework of the PSO algorithm is developed based on the following assumptions. The particles (solutions) fly randomly for the exploration of their environment (search space) and iteratively adjust their positions according to PSO operators to allocate the optimal solution (global best). The best positions located during the flying process toward the optimal position are stored. Other well-known swarm-based optimizer are Ant Colony Optimization (ACO) [19] and Artificial Bee Colony (ABC) [20]. Table 1 shows many others swarm optimization algorithms.

Physical-based algorithms are imitated by the physical phenomena that regulated and appeared in the universe. Many algorithms fall under this category, for example, Simulated Annealing (SA) which is inspired by the annealing process of metallurgy through putting the metal in heating followed by slow cooling to approach the best solution Physical-based algorithms are listed in Table 1. Other examples of algorithms that fall in this category are listed in Table 1.

Eventually, the human-based algorithm is another type of optimization algorithm that mimics human behavior and interactions in societies. An example of a human-based algorithm is the Harmony Search Algorithm (HSA), which is inspired by the music players' interactions with the notes of their instruments, they apply the best practices for approaching the desired harmony (optimal solution) [21]. The Fireworks algorithm is another example of human-based algorithms [22].

Conventionally, there is a bunch of nature-inspired algorithms which offer promising good solutions for a diverse scale of optimization problems. As mentioned before, there is no super optimization algorithm that can solve all classes of optimization problems efficiently [15]. Furthermore, optimization problems classified as non-linearity and multi-modality are arduous to be solved by deterministic algorithms. Therefore, researchers put great efforts into developing metaheuristic algorithms with diverse intelligence features from a wide variety of inspiration sources, to bring algorithms with robust optimization capabilities that can solve complex optimization problems successfully.

Table 1. Nature-inspired Metaheuristics.

Optimization Algorithms Type	Algorithms
Evaluation-based algorithms	Biogeography-Based Optimizer [12], Genetic Programming [10], Evolution Strategy [11], and Genetic Algorithm (GA) [9].
Chemical-Based algorithms	Chemical reaction optimisation [23].
Human-Based algorithms	β-Hill Climbing (βHC) [24], Coronavirus herd immunity optimizer (CHIO) [25], Fireworks algorithm [22], Group search optimizer [26], Harmony Search Algorithm (HSA) [21], Mine blast algorithm [27], Seeker optimization algorithm (SOA) [28], Social-based algorithm (SBA) [29], Tabu search (TS) [30], and Wisdom of artificial crowds (WAC) [31].
Physical-based algorithms	Big bang-big crunch (BBBC) [32], Charged system search (CSS) [33], Electromagnetism-like mechanism (EM) [34], Equilibrium optimizer (EO) [35], Gravitational search algorithm (GSA) [36], Henry gas solubility optimization (HGSO) [37], Water cycle algorithm (WCA) [38], Multi-verse optimizer (MVO) [39] and Sine cosine algorithm (SCA) [40].
Swarm-based algorithms	Ant colony optimization (ACO) [19], Ant lion optimizer (ALO) [41], Artifical bee colony (ABC) [20], Artificial fish-swarm algorithm (AFSA) [42], Bat algorithm (BA) [43], Bird mating optimizer (BMO) [44], Butterfly optimization algorithm (BOA) [45], Cat swarm optimization algorithm (CSOA) [46], Crow search algorithm (CSA) [47], Cuckoo search (CS) [48], Chicken swarm optimization (CSO) [49], Dragonfly algorithm (DA) [50], Elephant search algorithm (ESA) [51], Firefly algorithm [52], Flower pollination algorithm (FPA) [53], Salp Swarm Algorithm (SSA) [54], Moth-flame optimization algorithm (MFO) [55], Monarch butterfly optimization (MBO) [56], Grey wolf optimizer (GWO) [57], Fruit fly optimization algorithm (FOA) [58], Glowworm swarm optimization [59], Harris hawks optimization [60], Krill herd algorithm (KHA) [61], PSO [14], Red deer algorithm [62], Pelican optimization algorithm [63], Enhanced marine predators algorithm (LEO-MPA) [64] and Whale optimization algorithm (WOA) [65].

3. Lemurs Optimizer (LO)

In this section, the inspiration for the LO algorithm is first presented. After that, the mathematical model and the LO algorithm are discussed in detail.

3.1. Inspiration

Lemurs are classified as prosimian primates, which includes all primates that are neither monkeys nor apes [66]. Lemurs come in a diversity of varieties, but there are just a few individuals of each species. Just a small portion of the world is home to these primates. Many species have small populations that are dwindling. Lemurs can only be found in Madagascar and the neighboring Comoro Islands, which are located off the coast of Mozambique, Africa. They live in a variety of environments: mountains, wetlands, rain

forests, spiny forests, and dry deciduous forests. The indri is the largest lemur species. It can reach a weight of 15.5 to 22 pounds (7 to 10 kg) and a length of 24 to 35 inches (60 to 90 cm). Madame Berthe's mouse lemur is the tiniest of the lemurs, measuring 3.5 to 4 inches (9 to 11 cm) in length (not including the tail) [67].

Lemurs are highly social animals that live in groups known as troops. According to National Geographic, the ring-tail lemur's troop is led by a dominant female and can consist of six to thirty species. The majority of lemurs spend their waking hours in trees. Lemurs groom each other while they aren't feeding. The Lemurs communicate in two different ways. They communicate by vocalization and scent markings. Lemurs interact by emitting low growls. Sometimes it is a warning to flee, and sometimes a warming welcome. Soft purrs are used by mothers to communicate with their offspring. This also aids in the formation of strong bonds.

The pitch of a Lemur's shrill scream is extremely high. This is a warning signal that can be received from a long distance. This may be a territorial symbol, warning other Lemurs to stay away. Other times, it's a way of alerting the family that they are in danger and should seek shelter.

Lemurs have been observed meowing like cats. This form of sound is used to summon the family to a central position or to flee from predators such as fossa (the risk is very high). If they have spread out to search for food, this might be a way to get them all together for nesting.

Lemurs use their scent glands to convey their location. To locate food, the family groups may disperse. This may also assist dominant females in determining whether or not an alien has entered their family group and poses a challenge.

Lemurs have a wide range of locomotor behavior. For the LO algorithm, we used two main lemur behaviors as inspiration: leap up and dance-hup. In a leap up, the lemurs jump into the air and sit upright on a nearby branch, both hands and feet grasping the trunk closely. They have the potential to jump up to 10 m (33 ft) from tree trunk to tree trunk in a matter of seconds. The dance-hup occurs when the space between trees becomes too great, lemurs will descend to the ground and cross lengths of more than 100 m (330 feet) by standing upright and jumping horizontally with arms extended to the side and waving up and down from chest to head height, ostensibly for balance [68].

Figure 3 illustrates conceptual models of these two key lemur locomotor behaviors. The two stages of optimization using metaheuristics, exploration and exploitation, are very similar to these two Locomotor behaviors. The primary objective of the exploration phase is for lemurs to leap up over various areas to locate the best lemur location in the search space. However, lemurs in the dance-hub move into the best nearby lemur location and in one direction, which is useful during the exploitation phase. The following subsection will explain how the LO algorithm works conceptually and mathematically.

3.2. Mathematical Model of the Lemur Optimizer Algorithm

The search process is divided into two phases in the population-based algorithm, as described in the previous section: exploration versus exploitation. In the exploration phase, we utilize the dance-hup behavior. The leap-up behavior, on the other hand, aids LO in exploiting the search space. We consider each solution to be a lemur, with each vector representing a single one of the lemur's coordinates. We also allocate the best location to each solution that is related to the solution's fitness function value. As a result, the lemurs will change their place vectors and dance-hup towards the best neatest lemur or leap up to the global best lemur. Figure 4 illustrates the conceptual model of dance-hup and leap-up for the proposed algorithm.

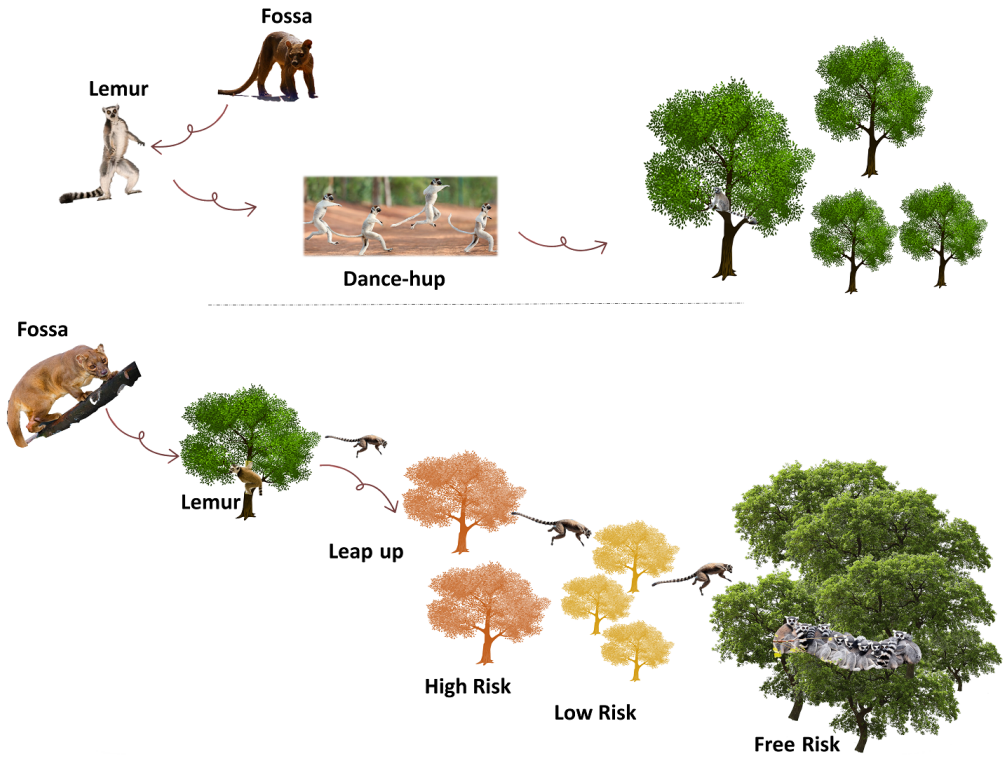


Figure 3. Lemur Optimizer Inspiration.

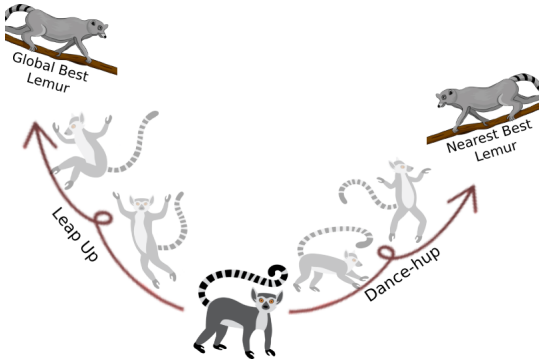


Figure 4. Leap Up and Dance Hub.

The set of lemurs is represented in a matrix since the LO algorithm is a population-based algorithm. To do this, the following procedures are carried out. Assuming that we have the population defined as the following matrix:

$$T = \begin{bmatrix} l_1^1 & l_1^2 & \cdots & l_1^d \\ l_2^1 & l_2^2 & \cdots & l_2^d \\ \vdots & \vdots & \vdots & \vdots \\ l_s^1 & l_s^2 & \cdots & l_s^d \end{bmatrix}. \quad (1)$$

where T denotes the set of lemurs in a population matrix of size $s \times d$, d denotes the decision variables, and s denotes the candidate solutions.

Typically, the decision variable j in solution i is randomly generated as follows:

$$l_i^j = rand() \times (ub_j - lb_j) + lb) \quad \forall i \in (1, 2, \ldots, n) \wedge \forall j \in (1, 2, \ldots, d) \tag{2}$$

where the function $rand()$ produces a random distributed number in the range $(1, 2, \ldots, MAX\_INT)$, MAX_INT, where $MAXINT$ is the largest integer number that can be generated, and the discrete lower and upper bound limits of variable j are denoted by $[lb_j, ub_j]$.

The lemur that has a lower fitness value tends to change its decision variables from the lemur that has a higher fitness value. This means that the overall fitness values of the total lemurs improve with iterations. Lemurs are organized based on their fitness values in each iteration, with one chosen as the global best lemur (i.e., gbl) and one chosen as the best nearest lemur for each lemur (i.e., bnl).

In this direction, the decision variable j in the solution i is assigned a value each iteration using two options: (a) the value is selected from the global best lemur, and (b) the value is selected from the best nearest lemur. This is formulated as shown in Equation (3).

$$L_i^j = \begin{cases} l(i,j) + abs(l(i,j) - l(bnl,j)) * (rand - 0.5) * 2; & rand < FRR, \\ l(i,j) + abs(l(i,j) - l(gbl,j)) * (rand - 0.5) * 2; & rand > FRR, \end{cases} \tag{3}$$

where $l(i,j)$ indicates j value of the current lemur, $l(bnl,j)$ indicates j value of the best nearest lemur for the the current lemur $l(i,j)$, $l(gbl,j)$ indicates the global best lemur, free risk rate (FRR) indicates the risk rate of the all lemurs in the troops, and $rand$ represents random numbers between [0, 1]. Based on this formulation, it can be concluded that the probability of the FRR is the main coefficient of the LO algorithm. The formula of this coefficient is given in:

$$FRR = FRR(High\_Risk\_Rate) - CurrIter \times ((High\_Risk\_Rate - Low\_Risk\_Rate)/MaxIter) \tag{4}$$

where Low_Risk_Rate and $High\_Risk\_Rate$ represent constant pre-defined values, $MaxIter$ is the maximum iterations' number, and $CurrIter$ denotes current iteration. Figure 5 illustrates the conceptual model of High-Risk Rate and Low-Risk Rate for the proposed algorithm. Note that the purpose of Low_Risk_Rate and $High\_Risk\_Rate$ is to determine the minimum and the maximum value for FRR.

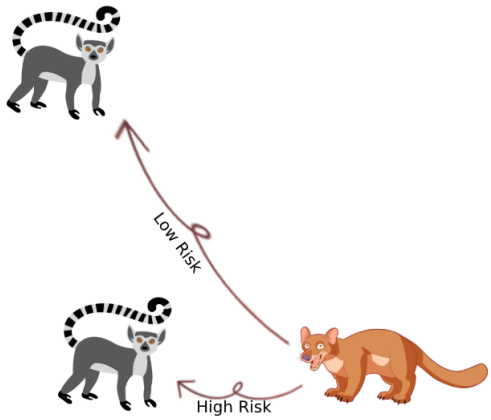


Figure 5. High-Risk Rate and Low-Risk Rate.

Algorithm 1 shows the LO algorithm's pseudocode, while the general steps are presented in Figure 6.

Algorithm 1 The LO algorithm's pseudocode.

Set up the LO parameters (Number of iterations, Number of dimensions (Dim), Number of solutions, Lower Bound (LB), Upper Bound (UB), Low-Risk Rate, High-Risk Rate).
Generate Lemurs population.
while the current iteration does not equal the number of iterations **do**
  Evaluate the objective function for all Lemurs.
  Calculate free risk rate (FRR) using Equation (4).
  Update the Global Best Lemur (gbl).
  for each lemur indexed by i **do**
    Update the Best Nearest Lemur (bnl).
    for each decision variable in Lemur i indexed by j **do**
      Set $random([0,1])$ to $rand$.
      if $rand < JumpingRrate$ **then**
        Use Equation (3) case number one to update the decision variable j.
      else
        Use Equation (3) case number two to update the decision variable j.
      end if

    end for
  end for
end while
Return the Global Best Lemur.

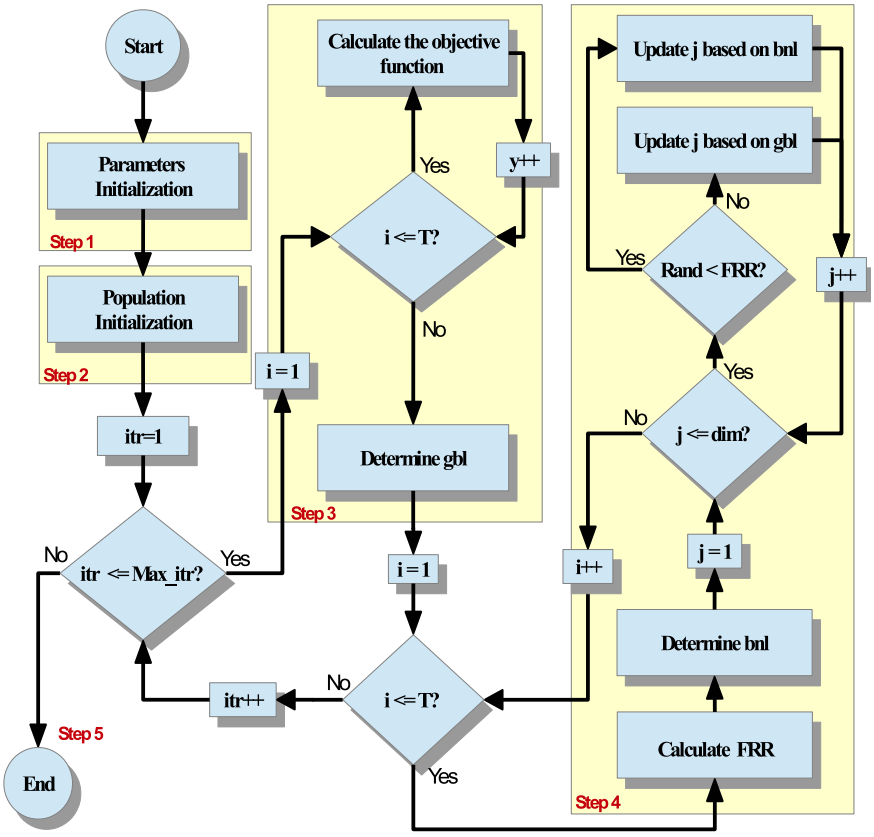


Figure 6. Lemur Optimizer Flowchart.

The LO algorithm begins by generating a swarm of lemurs randomly. At each iteration, the decision variables of the neatest lemur with a better fitness value try to transfer towards the lemurs with a lower fitness value via dance-hup.

In the LO algorithm, creating a set of random lemurs is the first step in the optimization process. The FRR value starts close to Low_Risk_Rate, which means the lemur tends to move toward the best neatest lemur via dance-hup. During the LO execution, the FRR will decrease close to $High\_Risk\_Rate$, which means the lemur tends to move toward the best global lemur via leap up. This procedure shall be iterated until an end condition is met.

The number of lemurs, iterations, and decision variables affects the algorithms' computing complexity. To summarise, the proposed algorithm has the following computational complexity:

$$O(LO) = O(MaxIter \times s \times d) \quad (5)$$

4. Experiments and Results

The proposed LO's performance is evaluated using 23 standard test functions in this section. Minimization optimization problems with various complexity and dimensional search spaces are used in these test cases. These test functions are divided into three categories, including unimodal, multimodal, and fixed-dimension multimodal functions. The primary features of the three categories are presented in Tables 2–4. The tables show the function name, the function mathematical model, the range of the search space's boundaries, the functions dimension (n), and the functions' optimum solution $f(x^*)$. The programming language MATLAB version 9.12.0 is used to conduct the experiments, and the code is available in the "Lemurs-Optimizer" GitHub, available online: https://github.com/ammarabbasi/Lemurs-Optimizer, accessed on 28 September 2022.

As mentioned previously, the test functions used in this section are divided into unimodal, multimodal, and fixed-dimension multimodal functions. The multimodal and fixed-dimension multimodal categories are similar but differ from each other to define the number of decision variables. The fixed-dimensional test functions provide various search spaces compared with multimodal test functions. However, tuning the decision variables cannot be done using the fixed-dimensional test functions' mathematical model. In this evaluation section, the unimodal functions are utilized to investigate the proposed LO exploitation ability, whereas the multimodal functions are utilized to examine and evaluate the exploration side of the LO [37].

4.1. Comparative Analysis with the Swarm-Based Optimization Algorithms

The LO is compared with six robust optimization algorithms, including ABC, SSA, SCA, BA, FPA, and JAYA to investigate and prove the proposed LO's robust performance. The population size and number of iterations used for all compared algorithms are 30 and 100,000, respectively. In addition, the low-scal0E-rate and high-scal0E-rate in LO are set to be 0.5 and 0.7, respectively.

Table 2. The characteristics of unimodal benchmark functions.

Function	Test Functions	Range	n	C	$f(x^*)$
F1	$\sum_{i=1}^{n} x_i^2$	$x_i \in [-100,100]$	30	U	0
F2	$\sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	$x_i \in [-10,10]$	30	U	0
F3	$\sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j \right)^2$	$x_i \in [-100,100]$	30	U	0
F4	$\max_i \{ \lvert x_i \rvert, 1 \leq i \leq n \}$	$x_i \in [-100,100]$	30	U	0
F5	$\sum_{i=1}^{n-1} \left[100 \left(x_{i+1} - x_i^2 \right)^2 + (x_i - 1)^2 \right]$	$x_i \in [-30,30]$	30	U	0
F6	$\sum_{i=1}^{n} (\lvert x_i + 0.5 \rvert)^2$	$x_i \in [-100,100]$	30	U	0
F7	$\sum_{i=1}^{n} i x_i^4 + random[0,1)$	$x_i \in [-128,128]$	30	U	0

Table 3. The characteristics of multimodal benchmark functions.

Function	Test Functions	Range	n	C	$f(x^*)$
F8	$\sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	$x_i \in [-500,500]$	30	M	−12,569.5
F9	$\sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$	$x_i \in [-5.12,5.12]$	30	M	0
F10	$-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp\left(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)\right) + 20 + e$	$x_i \in [-32,32]$	30	M	0
F11	$\frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$x_i \in [-600,600]$	30	M	0
F12	$\frac{\pi}{n}\left\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i-1)^2\left[1+10\sin^2(\pi y_{i+1})\right] + (y_n-1)^2\right\} + \sum_{i=1}^{n} u(x_i,10,100,4)$ $y_i = 1 + \frac{x_i+1}{4}$ $u(x_i,a,k,m) = \begin{cases} k(x_i-a)^m & x_i > a \\ 0-a & < x_i < a \\ k(-x_i-a)^m & x_i < -a \end{cases}$	$x_i \in [-50,50]$	30	M	0
F13	$0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i-1)^2\left[1+\sin^2(3\pi x_i+1)\right] + (x_n-1)^2\left[1+\sin^2(2\pi x_n)\right]\right\} + \sum_{i=1}^{n} u(x_i,5,100,4)$	$x_i \in [-50,50]$	30	M	0

Table 4. The characteristics of fixed-dimension multimodal benchmark functions.

Function	Test Functions	Range	n	C	$f(x^*)$
F14	$\left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6}\right)^{-1}$	$x_i \in [-65,65]$	2	M	1
F15	$\sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2+b_i x_2)}{b_i^2+b_i x_3+x_4}\right]^2$	$x_i \in [-5,5]$	4	M	0.00030
F16	$4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	$x_i \in [-5,5]$	2	M	−1.0316
F17	$\left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	$x_i \in [-5,5]$	2	M	0.398
F18	$\left[1 + (x_1+x_2+1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right]$ $\times\left[30 + (2x_1-3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\right]$	$x_i \in [-2,2]$	2	M	3
F19	$-\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right)$	$x_i \in [1,3]$	3	M	−3.86
F20	$-\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2\right)$	$x_i \in [0,1]$	6	M	−3.32
F21	$-\sum_{i=1}^{5}\left[(X-a_i)(X-a_i)^T + c_i\right]^{-1}$	$x_i \in [0,10]$	4	M	−10.1532
F22	$-\sum_{i=1}^{7}\left[(X-a_i)(X-a_i)^T + c_i\right]^{-1}$	$x_i \in [0,10]$	4	M	−10.4028
F23	$-\sum_{i=1}^{10}\left[(X-a_i)(X-a_i)^T + c_i\right]^{-1}$	$x_i \in [0.10]$	4	M	−10.5363

4.2. Evaluation of Exploitation Capability (Functions F1–F7)

Given that one global optimum exists for each of the unimodal functions (F1–F7), they investigate the compared algorithms' exploitation capability. Table S1 demonstrates that the proposed LO shows high performance in optimizing the functions and reducing their values, and achieving the best exploitation capability. Particularly, the proposed LO obtains the best results in optimizing F1 and F2 in terms of best, worst, and mean and gets the second-best results in most of the other functions (i.e., F3–F7). Accordingly, it is notable that the proposed LO achieves the best exploitation capability.

4.3. Evaluation of Exploration Capability (Functions F8–F23)

The multimodal functions have several local optimums determined by the problem size (i.e., decision variables), where the number of local optimums increases with increasing the size of the problem. Accordingly, the multimodal functions play the primary role in evaluating the optimization algorithm's exploration capability. The results presented in Table S2 prove the demonstration of the proposed LO against the compared algorithms, where the LO obtains the best results in achieving the best in ten functions, including F8, F10, F11, F14, F21, and F23, and Worst and Mean in 10 functions, including F10, F14, F16–F19, and F23. These results prove the LO's robust performance in managing its exploration capability, which significantly leads LO towards the global optimum. The detailed results can be found in the Supplemental Information in Tables S1 and S2.

4.4. Analysis of Convergence Behavior

In the optimization process of LO, the search agents share their information and exploit the best lemur to scan the search space effectively to reach the most promising regions in the search space. The search agents in the early stage of optimization allocate abruptly positions and then gradually converge. Researchers in [69] stated such behavior in other population-based algorithms can lead to achieving the desired convergence. Convergence curves of LO, ABC, SSA, BAT, FPA, and JAYA are plotted in Figure 7 based on the average best-so-far in each iteration over 30 runs for some of the unimodal and multimodal benchmark functions in this study. It can be observed that the convergence trend of LO is competitive with other stat0E-of-th0E-art meta-heuristic algorithms.

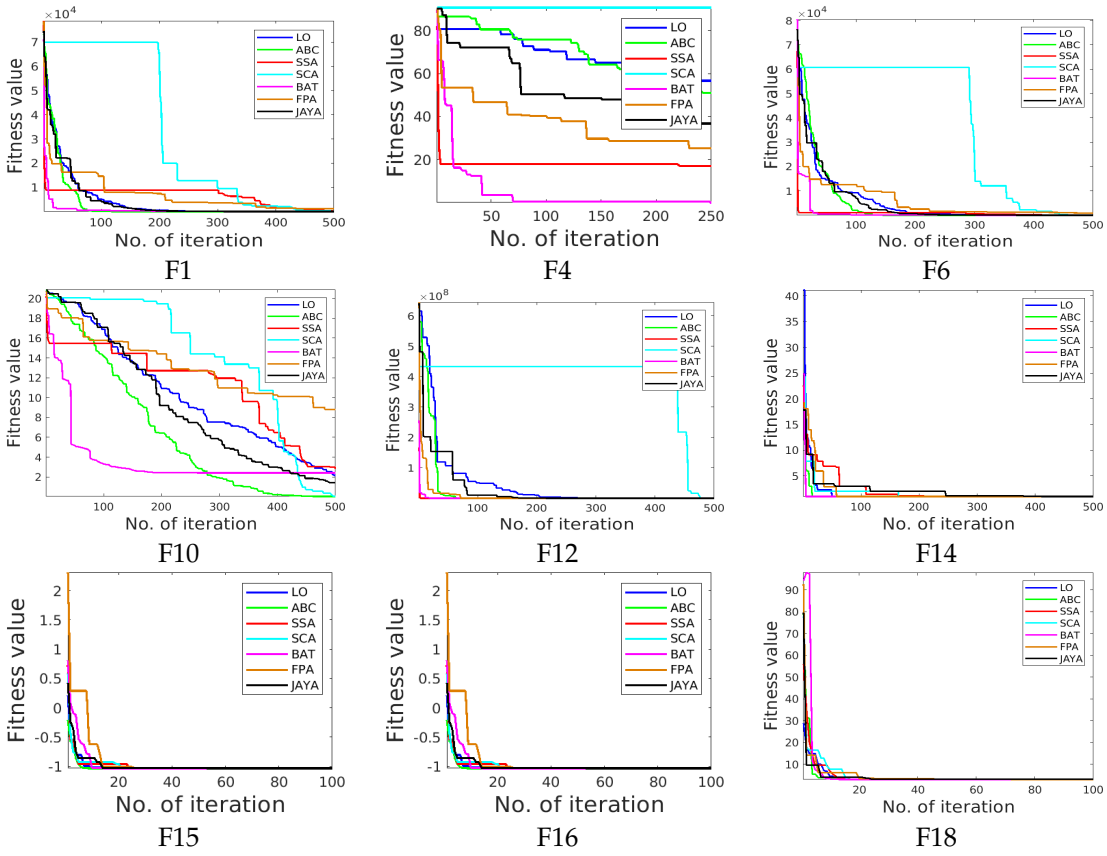


Figure 7. LO algorithm convergence plots with the other swarm−based algorithms.

Moreover, the convergence rate of LO based on best and average fitness is plotted in Figures 8 and 9. There is a descending pattern in the average fitness throughout the growth of the iterations in these figures' second and fourth columns. The successful convergence of LO in optimizing benchmark functions is owing to its search capabilities in terms of leap up (exploitation) and dance hub (exploration) that attractively improve the trajectory of lemurs toward optimality.

Further, the convergence curves in Figures 8 and 9 demonstrate that the LO algorithm produces better quality solutions throughout the optimization iterations.

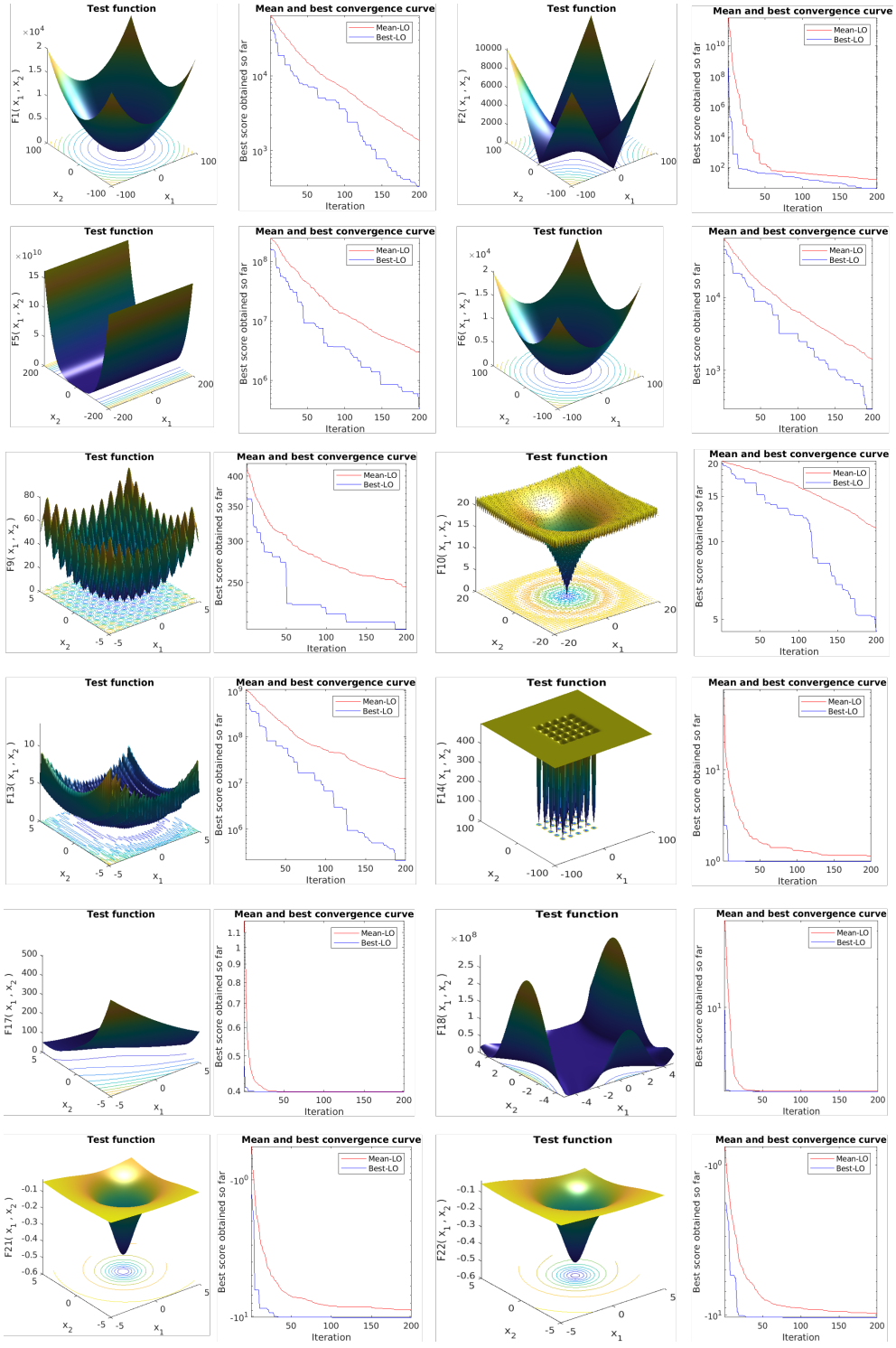


Figure 8. Functions and convergence plots.

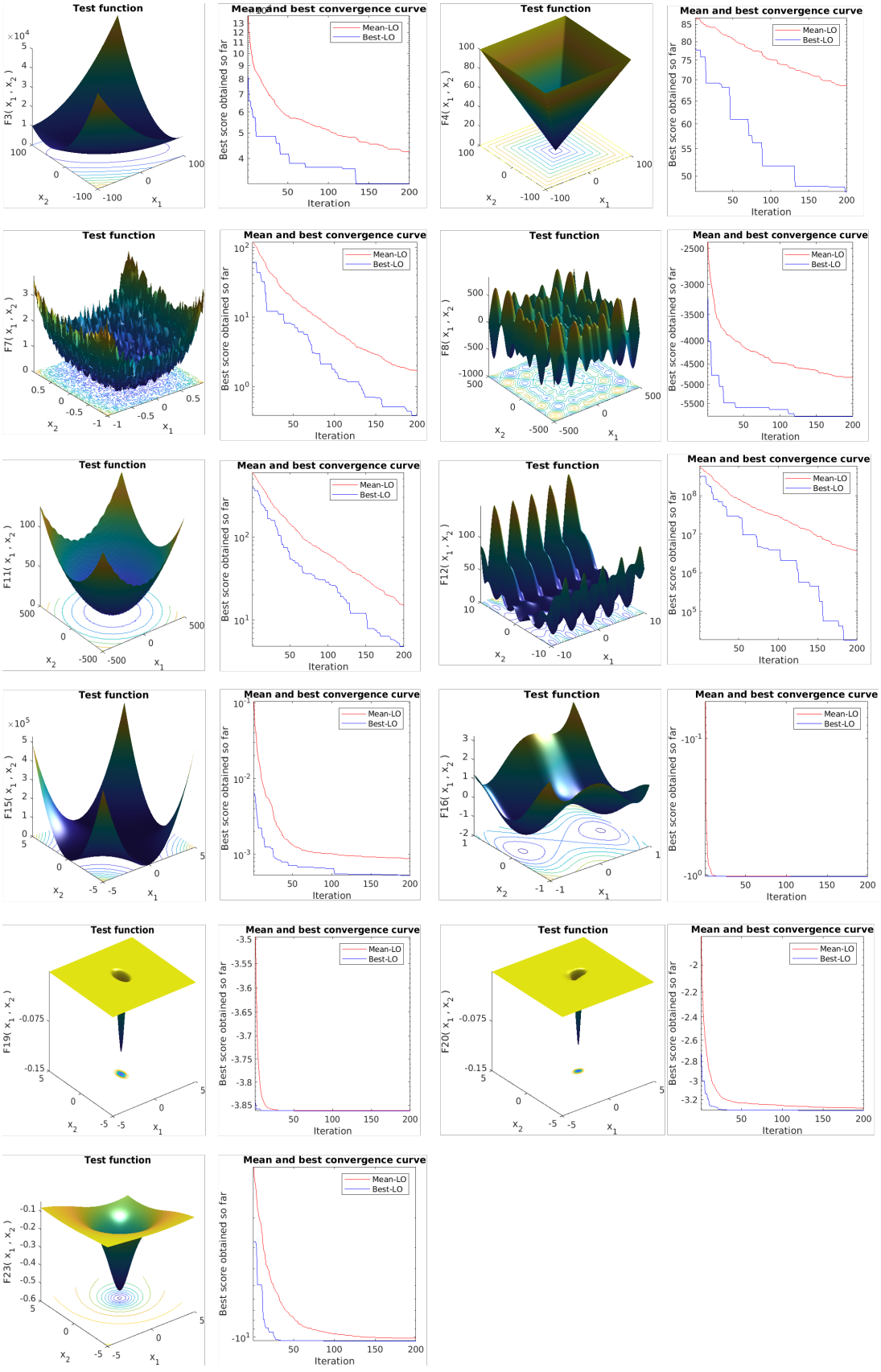


Figure 9. Functions and convergence plots.

Figure 10 illustrates the average rankings of the proposed algorithm and other comparative algorithms according to the average of the results. These rankings are calculated using Friedman's test. It should be noted that the lower value of the rankings, the better performance. It can be seen that the LEO-MPA is placed first by getting the lowest rankings, while the proposed LO is ranked third.

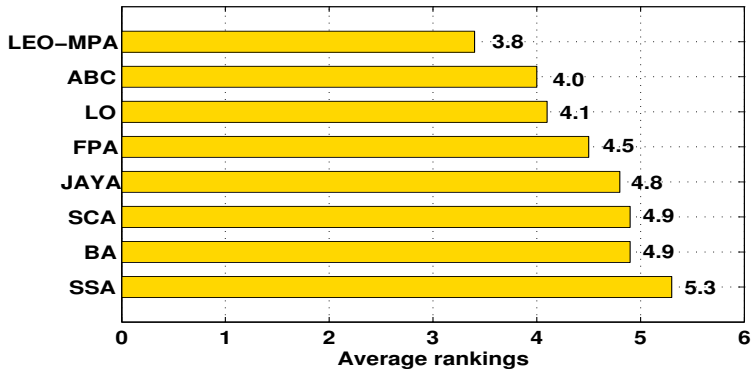


Figure 10. Average rankings of all comparative methods on all test functions.

By following [55], the signed Wilcoxon statistical test [70] was used to determine if there is a significant improvement between the LO algorithm and the other comparative algorithms. For each algorithm, the best results of 30 runs are used in the Wilcoxon signed rank with p value equal to 0.05. The proposed LO's improvements were tested to see whether they happened by chance, or were statistically significant using this test. The p value was determined using the statistically signed Wilcoxon statistical test. Two hypotheses, the null hypothesis, and the alternative hypothesis were considered in our experiment. The null hypothesis states that the mean values of the LO and other algorithms do not vary significantly (i.e., "−"). The alternative hypothesis, on the other hand, revealed that the mean values of the LO and other algorithms (i.e., "+") differ significantly. All other algorithms and the LO pair-wise are shown in Tables 5 and 6, indicating whether the null or alternative hypothesis is accepted. The proposed algorithm outperformed other algorithms by the smallest p value. The detailed results can be found in the Supplemental Information in Table S3.

Table 5. Comparative performance of the LO algorithm with swarm algorithms based on functions F1–F7.

Function	LO	ABC	Significantly	SSA	Significantly	SCA	Significantly	BA	Significantly
F1	$\mathbf{0.0000 \times 10^{0}}$	2.3350×10^{-16}	+	3.6020×10^{-10}	+	$\mathbf{0.0000 \times 10^{0}}$	–	5.8240×10^{-6}	+
F2	$\mathbf{0.0000 \times 10^{0}}$	9.5130×10^{-16}	+	1.4180×10^{-6}	+	$\mathbf{0.0000 \times 10^{0}}$	–	1.0390×10^{-1}	+
F3	1.3700×10^{-5}	1.3010×10^{0}	+	3.1910×10^{-11}	+	1.9590×10^{-29}	+	1.9530×10^{-3}	+
F4	5.1440×10^{-14}	1.4670×10^{-2}	+	2.9060×10^{-6}	+	1.3770×10^{-46}	+	1.3340×10^{-2}	+
F5	3.4140×10^{1}	8.2800×10^{-3}	+	9.1380×10^{0}	+	2.6760×10^{1}	+	3.0930×10^{-1}	+
F6	9.7010×10^{-28}	3.3340×10^{-16}	+	2.8460×10^{-11}	+	2.7520×10^{0}	+	9.6080×10^{-4}	+
F7	$\mathbf{5.8730 \times 10^{-6}}$	2.7370×10^{-2}	+	6.0940×10^{-5}	–	1.9790×10^{-4}	–	2.8460×10^{-4}	+
		FPA		JAYA		LEO-MPA			
F1		1.2930×10^{-68}	+	$\mathbf{0.0000 \times 10^{0}}$	–	$\mathbf{0.0000 \times 10^{0}}$	–		
F2		3.2560×10^{-47}	+	$\mathbf{0.0000 \times 10^{0}}$	+	$\mathbf{0.0000 \times 10^{0}}$	+		
F3		7.0320×10^{-31}	+	6.1680×10^{0}	+	$\mathbf{0.0000 \times 10^{0}}$	+		
F4		6.1330×10^{0}	+	$\mathbf{8.5570 \times 10^{-75}}$	–	8.5256×10^{-49}	–		
F5		1.0630×10^{0}	+	$\mathbf{7.1060 \times 10^{-28}}$	–	6.6720×10^{-22}	–		
F6		$\mathbf{1.0270 \times 10^{-33}}$	–	1.7170×10^{0}	+	1.3482×10^{0}	+		
F7		7.0020×10^{-3}	+	8.0060×10^{-4}	+	2.8813×10^{-4}	+		

Table 6. Comparative performance of the LO algorithm with swarm algorithms based on functions F8–F23.

Function	LO	ABC	Significantly	SSA	Significantly	SCA	Significantly	BA	Significantly
F8	-1.1930×10^{4}	$\mathbf{-1.2570 \times 10^{4}}$	+	-3.3000×10^{3}	+	-4.8300×10^{3}	+	$\mathbf{-1.2570 \times 10^{4}}$	–
F9	3.0340×10^{1}	$\mathbf{0.0000 \times 10^{0}}$	–	1.1410×10^{1}	+	$\mathbf{0.0000 \times 10^{0}}$	–	3.4020×10^{-2}	+
F10	6.8090×10^{-15}	2.8480×10^{-14}	+	2.1420×10^{-6}	+	4.7440×10^{0}	+	2.2490×10^{-2}	+
F11	$\mathbf{0.0000 \times 10^{0}}$	$\mathbf{0.0000 \times 10^{0}}$	+	2.4430×10^{-1}	+	$\mathbf{0.0000 \times 10^{0}}$	–	3.8230×10^{-3}	+
F12	$\mathbf{6.3590 \times 10^{-30}}$	3.0070×10^{-16}	+	2.0440×10^{-13}	+	2.3450×10^{-1}	+	8.0380×10^{-6}	+
F13	2.8610×10^{-24}	2.9820×10^{-16}	+	1.1170×10^{-12}	+	1.7140×10^{0}	+	1.2240×10^{-3}	+
F14	$\mathbf{9.9800 \times 10^{-1}}$	$\mathbf{9.9800 \times 10^{-1}}$	+	$\mathbf{9.9800 \times 10^{-1}}$	+	$\mathbf{9.9800 \times 10^{-1}}$	+	$\mathbf{9.9800 \times 10^{-1}}$	+
F15	3.6590×10^{-4}	3.4470×10^{-4}	+	5.5170×10^{-4}	+	3.0990×10^{-4}	+	$\mathbf{3.0750 \times 10^{-4}}$	+
F16	$\mathbf{-1.0320 \times 10^{0}}$	$\mathbf{-1.0320 \times 10^{0}}$	+	$\mathbf{-1.0320 \times 10^{0}}$	+	$\mathbf{-1.0320 \times 10^{0}}$	+	$\mathbf{-1.0320 \times 10^{0}}$	+
F17	3.9790×10^{-1}	3.9790×10^{-1}	+	3.9790×10^{-1}	+	3.9790×10^{-1}	+	3.9790×10^{-1}	+
F18	$\mathbf{3.0000 \times 10^{0}}$	$\mathbf{3.0000 \times 10^{0}}$	+	$\mathbf{3.0000 \times 10^{0}}$	+	$\mathbf{3.0000 \times 10^{0}}$	+	$\mathbf{3.0000 \times 10^{0}}$	+
F19	$\mathbf{-3.8630 \times 10^{0}}$	$\mathbf{-3.8630 \times 10^{0}}$	+	$\mathbf{-3.8630 \times 10^{0}}$	+	-3.8560×10^{0}	+	$\mathbf{-3.8630 \times 10^{0}}$	+
F20	-3.3180×10^{0}	$\mathbf{-3.3220 \times 10^{0}}$	+	-3.2150×10^{0}	+	-2.9250×10^{0}	+	-3.2820×10^{0}	+
F21	-9.9040×10^{0}	-1.0150×10^{1}	+	-1.0150×10^{1}	+	-3.3210×10^{0}	+	-1.0150×10^{1}	+
F22	-1.0400×10^{1}	-1.0400×10^{1}	+	-1.0400×10^{1}	+	-5.3910×10^{1}	+	-1.0150×10^{1}	+
F23	-1.0540×10^{1}	-1.0540×10^{0}	+	-1.0000×10^{1}	+	-6.2120×10^{1}	+	-1.0150×10^{1}	+
		FPA		JAYA		LEO-MPA			
F8		-1.2530×10^{4}	+	-1.2410×10^{4}	+	-1.2289×10^{4}	+		
F9		1.7350×10^{1}	+	4.1040×10^{1}	+	$\mathbf{0.0000 \times 10^{0}}$	+		
F10		2.0940×10^{0}	+	1.0480×10^{-14}	+	$\mathbf{4.4409 \times 10^{-16}}$	+		
F11		2.2510×10^{-2}	+	9.6030×10^{-3}	+	$\mathbf{0.0000 \times 10^{0}}$	–		
F12		3.1100×10^{-2}	+	8.0000×10^{-1}	+	1.5705×10^{-29}	+		
F13		2.1660×10^{-3}	+	1.0990×10^{-3}	+	$\mathbf{1.3498 \times 10^{-32}}$	+		
F14		$\mathbf{9.9800 \times 10^{-1}}$	+	$\mathbf{9.9800 \times 10^{-1}}$	+	$\mathbf{9.9800 \times 10^{-1}}$	–		
F15		$\mathbf{3.0750 \times 10^{-4}}$	+	3.3800×10^{-4}	+	3.0789×10^{-4}	+		
F16		$\mathbf{-1.0320 \times 10^{0}}$	+	$\mathbf{-1.0320 \times 10^{0}}$	+	-1.0316×10^{0}	–		
F17		3.9790×10^{-1}	+	3.9890×10^{-1}	+	$\mathbf{3.9789 \times 10^{-1}}$	+		
F18		$\mathbf{3.0000 \times 10^{0}}$	+	$\mathbf{3.0000 \times 10^{0}}$	+	$\mathbf{3.0000 \times 10^{0}}$	+		
F19		$\mathbf{-3.8630 \times 10^{0}}$	+	$\mathbf{-3.8630 \times 10^{0}}$	+	-3.8628×10^{0}	+		
F20		$\mathbf{-3.3220 \times 10^{0}}$	+	-3.2670×10^{0}	+	-3.3061×10^{0}	+		
F21		-1.0150×10^{1}	+	-8.1430×10^{0}	+	$\mathbf{-1.0153 \times 10^{1}}$	+		
F22		-1.0400×10^{1}	+	$\mathbf{-8.5510 \times 10^{1}}$	+	-1.0626×10^{1}	+		
F23		-1.0540×10^{1}	+	$\mathbf{-9.9950 \times 10^{1}}$	+	-1.0636×10^{1}	+		

Figure 11 shows the time-average rankings of the proposed lo and other comparative algorithms according to the average time of all 30 runs. This figure clearly shows that the proposed LEO-MPA takes more time to finish than other algorithms. In contrast, the proposed LO is ranked first.

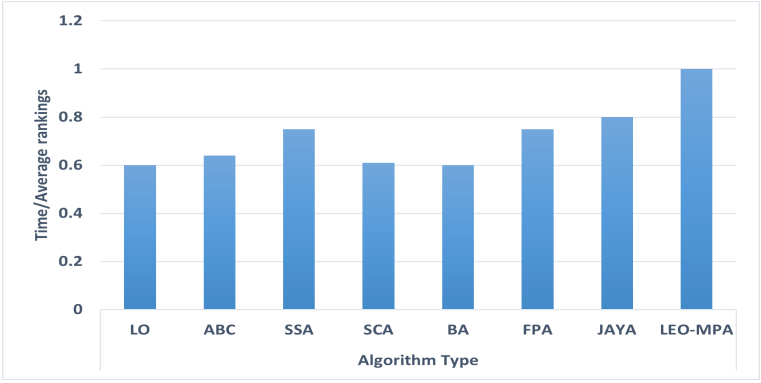


Figure 11. Time average rankings of all comparative methods on all test functions.

The results of this section revealed various characteristics of the proposed LO algorithm. The location updating mechanism of lemurs using Equation (3) case number one is responsible for LO's high exploration ability. During the initial steps of the iterations, this equation allows lemurs to move around the best nearest lemur. The rest of the iterations emphasize high exploitation and convergence, which come from Equation (3) case number two. This equation enables the lemurs to quickly reposition themselves around or step towards the current global best lemur. It is worth mentioning here that the exploration and exploitation phases are completed separately, and the LO exhibits convergence speed and high local optima avoidance at the same time. Besides, the LO is utilized one formula to manage these two phases and update the position of lemurs. The performance of LO in real engineering problems is verified in the following section.

4.5. Engineering Optimization Problems in the Real World

In this section, three well-known real-world problems, presented at (IEE0E-CEC 2011) the 2011 IEEE Congress on Evolutionary Computation, are addressed to evaluate the efficiency of the proposed LO algorithm [71]. In this regard, transmission network expansion planning (TNEP), the bifunctional catalyst blend optimal control (BCBOC), and Parameter estimation for frequency-modulated (FM) sound waves (PEFMSW) problems are utilized. Table 7 shows the characteristics of these real-world problems in terms of the problem dimension and the value range of decision variables.

Table 7. The characteristics of three real-world problems.

Problem	Value Range	Dimension
PEFMSW	$x_i \in [-6.4, 5.35]$,where $1 \leq i \leq 6$	6
TNEP	$x_1, x_2 \in [0, 4]$, $x_3 \in [0, \pi]$, $x_i \in [-4 - \frac{1}{4}\lfloor\frac{i-4}{3}\rfloor, 4 + \frac{1}{4}\lfloor\frac{i-4}{3}\rfloor]$, where $4 \leq i \leq 30$	30
BCBOC	$x_1 \in [0.6, 0.9]$	1

It should be noted that the parameter settings for the proposed LO algorithm are as follow: the number of runs is 30, the number of iterations is 150,000 and the population size is 30. These settings are under the IEE0E-CEC2011 rules [71]. These settings are suggested to make a fair comparison with thirteen other comparative algorithms. These comparative algorithms include CHIO (i.e., coronavirus herd immunity optimizer) [25], APS (i.e., adaptive population-based simplex algorithm) [72], ADE (i.e., adaptive differential evolution algorithm) [73], CDASA (i.e., continuous differential ant-stigmergy Algorithm) [74], DE (i.e., differential evolution) [75], D0E-RHC [69], GA-MPC [76], HDE [77], HMA (i.e., hybrid EA-D0E-memetic algorithm) [78], IMO (i.e., intellects-masses optimizer) [79], ABC (i.e., artificial bee colony) [80], AABC (i.e., accelerated artificial Bee colony algorithm) [80], and KHABC [81].

4.6. Transmission Network Expansion Planning (TNEP) Problem

The TNEP problem entails finding the lowest cost transmission assets that can be installed in a power system to meet predicted demand over a specified time horizon [82]. Because TNEP has a long-term effect on system operation, it is a key strategic decision in power systems. Additionally, TNEP is a non-linear, non-convex, and multi-modal optimization problem that is classified as NP-hard in terms of computing complexity. Different models and strategies for solving the TNEP problem have been developed in the existing literature. To address the TNEP problem in its various manifestations, heuristic and metaheuristic have been developed. While heuristic techniques are simple to use, they typically become stuck in locally optimal solutions. Metaheuristic techniques are more efficient search algorithms that are capable of finding better solutions than conventional heuristic techniques at the cost of increased processing time. Table 8 compares the performance of the proposed LO algorithm with eleven different comparison algorithms. In this table, the experimental results of each comparative algorithm are summarized in terms of the best, mean, worst, median, and standard derivations across 30 runs. From Table 8, it is clear that the proposed LO algorithm performance is similar to all other algorithms by obtaining the same results (i.e., it reached the optimal solution).

Table 8. The performance of the LO algorithm against other comparative algorithms on transmission network expansion planning problems.

Algorithm	Best	Mean	Median	Worst	Stdv
LO	$\mathbf{0.0220000 \times 10^4}$	0.0220000×10^4	0.0220000×10^4	0.0220000×10^4	0.0220000×10^4
HMA	$\mathbf{0.0220000 \times 10^4}$	0.0220000×10^4	0.0220000×10^4	0.0220000×10^4	0.0220000×10^4
DE	$\mathbf{0.0220000 \times 10^4}$	0.0220000×10^4	0.0220000×10^4	0.0220000×10^4	0.0220000×10^4
APS 9	$\mathbf{0.0220000 \times 10^4}$	0.0220000×10^4	0.0220000×10^4	0.0220000×10^4	0.0220000×10^4
D0E-RHC	$\mathbf{0.0220000 \times 10^4}$	0.0220000×10^4	N/A	0.0220000×10^4	0.0220000×10^4
CHIO	$\mathbf{0.0220000 \times 10^4}$	0.0220000×10^4	0.0220000×10^4	0.0220000×10^4	0.0220000×10^4
HDE	$\mathbf{0.0220000 \times 10^4}$	0.0220000×10^4	0.0220000×10^4	0.0220000×10^4	0.0220000×10^4
ADE	$\mathbf{0.0220000 \times 10^4}$	0.0220000×10^4	0.0220000×10^4	0.0220000×10^4	0.0220000×10^4
IMO	$\mathbf{0.0220000 \times 10^4}$	0.0220000×10^4	0.0220000×10^4	0.0220000×10^4	0.0220000×10^4
KHABC	$\mathbf{0.0220000 \times 10^4}$	0.0220000×10^4	0.0220000×10^4	0.0220000×10^4	0.0220000×10^4
CDASA	$\mathbf{0.0220000 \times 10^4}$	0.0220000×10^4	0.0220000×10^4	1.43290×10^1	0.0220000×10^4
GA-MPC	$\mathbf{0.0220000 \times 10^4}$	0.0220000×10^4	0.0220000×10^4	0.0220000×10^4	0.0220000×10^4

4.7. The Bifunctional Catalyst Blend Optimal Control Problem

The experimental results of the proposed LO algorithm, as well as the results of ten of the comparative algorithms, are recorded in Table 9. It can be observed from the results in Table 9 that the performance of the LO algorithm is similar to other algorithms by obtaining the same best results.

Table 9. The performance of the LO algorithm against other comparative algorithms on the bifunctional catalyst blend optimal control problem.

Algorithm	Best	Mean	Median	Worst	Stdv
LO	$\mathbf{0.0115150 \times 10^{-3}}$	0.0115150×10^{-3}	0.0115150×10^{-3}	0.0115150×10^{-3}	4.53430×10^{-20}
HMA	$\mathbf{0.0115150 \times 10^{-3}}$	0.0115150×10^{-3}	0.0115150×10^{-3}	0.0115150×10^{-3}	9.97110×10^{-19}
DE	$\mathbf{0.0115150 \times 10^{-3}}$	0.0115150×10^{-3}	0.0115150×10^{-3}	0.0115150×10^{-3}	2.00390×10^{-19}
APS 9	$\mathbf{0.0115150 \times 10^{-3}}$	0.0115150×10^{-3}	0.0115150×10^{-3}	0.0115150×10^{-3}	4.80700×10^{-19}
D0E-RHC	$\mathbf{0.0115150 \times 10^{-3}}$	0.0115150×10^{-3}	N/A	0.0115150×10^{-3}	0.00000×10^{0}
CHIO	$\mathbf{0.0115150 \times 10^{-3}}$	0.0115150×10^{-3}	0.0115150×10^{-3}	0.0115150×10^{-3}	4.53430×10^{-10}
HDE	$\mathbf{0.0115150 \times 10^{-3}}$	0.0115150×10^{-3}	0.0115150×10^{-3}	0.0115150×10^{-3}	6.10870×10^{-18}
ADE	$\mathbf{0.0115150 \times 10^{-3}}$	0.0115150×10^{-3}	0.0115150×10^{-3}	0.0115150×10^{-3}	3.80430×10^{-19}
IMO	$\mathbf{0.0115150 \times 10^{-3}}$	0.0115150×10^{-3}	0.0115150×10^{-3}	0.0115150×10^{-3}	0.00000×10^{0}
CDASA	$\mathbf{0.0115150 \times 10^{-3}}$	0.0115150×10^{-3}	0.0115150×10^{-3}	0.0115150×10^{-3}	1.68850×10^{-24}
GA-MPC	$\mathbf{0.0115150 \times 10^{-3}}$	0.0115150×10^{-3}	0.0115150×10^{-3}	0.0115150×10^{-3}	0.00000×10^{0}

4.8. Parameter Estimation for Frequency-Modulated (FM) Sound Waves

The experimental results of running the proposed LO algorithm are recorded in Table 10. In the same table, these findings are compared to those of thirteen different comparative algorithms. From Table 10, it can be shown that the LO algorithm performs similarly to five of the other compared algorithms in terms of attaining optimal outcomes when solving such problems. Furthermore, the LO algorithm is obtained the same best results at all times of runs, and this is similar to the GA-MPC algorithm. Thus, the effectiveness of the proposed LO in handling complex optimization problems is demonstrated.

Table 10. The performance of the LO algorithm against other comparative algorithms on Parameter estimation for frequency-modulated (FM) sound waves.

Algorithm	Best	Mean	Median	Worst	Stdv
LO	$\mathbf{0.000000 \times 10^{0}}$	0.000000×10^{0}	0.000000×10^{0}	0.000000×10^{0}	0.000000×10^{0}
HMA	1.167400×10^{-11}	2.094900×10^{0}	6.084700×10^{-10}	1.137400×10^{1}	4.306400×10^{0}
DE	$\mathbf{0.000000 \times 10^{0}}$	6.044800×10^{-13}	1.085400×10^{-27}	1.312700×10^{-11}	2.638800×10^{-12}
APS 9	$\mathbf{0.000000 \times 10^{0}}$	1.193500×10^{1}	1.481300×10^{1}	1.869800×10^{1}	6.516900×10^{0}
D00E-RHC	5.020000×10^{-20}	8.910000×10^{0}	N/A	1.560000×10^{1}	6.370000×10^{0}
CHIO	9.057300×10^{0}	1.831100×10^{1}	1.706000×10^{1}	4.023300×10^{1}	8.153800×10^{0}
HDE	7.209300×10^{-15}	8.769700×10^{-1}	1.236200×10^{-11}	1.175700×10^{1}	3.043900×10^{0}
ADE	$\mathbf{0.000000 \times 10^{0}}$	3.852600×10^{0}	0.000000×10^{0}	1.702100×10^{1}	5.690000×10^{0}
IMO	$\mathbf{0.000000 \times 10^{0}}$	8.989400×10^{-1}	0.000000×10^{0}	1.230600×10^{1}	3.126600×10^{0}
CDASA	3.278900×10^{-18}	$0.01151500 \times 10^{-3}$	1.137600×10^{1}	2.117100×10^{1}	7.095500×10^{0}
GA-MPC	$\mathbf{0.000000 \times 10^{0}}$	0.000000×10^{0}	0.000000×10^{0}	0.000000×10^{0}	0.000000×10^{0}
KHABC	1.231000×10^{1}	2.231000×10^{1}	N/A	2.779000×10^{1}	3.530000×10^{0}
AABC	3.669600×10^{-1}	N/A	N/A	N/A	N/A
ABC	2.772500×10^{0}	N/A	N/A	N/A	N/A

All of the previous experiments and observations support the proposed algorithm's ability to solve complex problems with unknown search spaces. As a result, this efficient optimization algorithm is being provided to be utilized for optimization problems in various fields. It is worth mentioning that other well-defined optimization problems such as text documents clustering [83,84], EEG signals denoising [85–87], feature selection [88–91], and scheduling problems in smart home [92–95] can be handled by the proposed algorithm.

5. Conclusions and Future Works

In this study, an innovative evolutionary optimization technique inspired by the locomotor behavior of lemurs was developed. The proposed algorithm is called Lemur Optimizer (LO) which included one operator to simulate how lemurs escape from predator attacks and search for food. The proposed algorithm is investigated from different aspects which are: convergence behavior, exploitation, and exploration. The conclusion can be summarized as follows:

- An exhaustive analysis was performed on 23 mathematical benchmark functions. In contrast to other state-of-the-art metaheuristic algorithms, LO is found to be sufficiently competitive.
- Three structural engineering problems (i.e., Transmission network expansion planning (TNEP) problem, The bifunctional catalyst blend optimal control problem, and Parameter estimation for frequency-modulated (FM) sound waves) are studied and used to evaluate the performance of the proposed algorithm.
- The results show that LO is very competitive when compared to other metaheuristic algorithms.

As a future direction, we will develop a multi-objective version of the LO algorithm. Another future work can be introducing a binary version LO algorithm.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/app121910057/s1, Table S1: Comparative performance of the LO algorithm with swarm algorithms based on functions F1–F7; Table S2: Comparative performance of the LO algorithm with swarm algorithms based on functions F8–F23; Table S3: p_values of the Wilcoxon statistical test between the proposed LO algorithm and ABC, SCA, BA, FPA, JAYA, and LEO-MPA.

Author Contributions: A.K.A. proposed an idea and experimented. S.N.M., M.A.A.-B., O.A.A., M.A.A., Z.A.A.A., E.H.A. and I.A.D. discussed the idea and worked with A.K.A. for further implementation and experimentation. S.N.M., M.H., A.E. and M.A.A.-B., wrote the main sections manuscript, and all authors finalized the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by Taif University Researchers Supporting Project number (TURSP-2020/292) Taif University, Taif, Saudi Arabia. This work is also supported by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R193), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Informed Consent Statement: Hereby, we consciously assure that for the manuscript "Lemurs optimizer: A new metaheuristic algorithm for global optimization" the following is fulfilled: 1. This material is the authors' original work, which has not been previously published elsewhere. 2. The paper is not currently being considered for publication elsewhere. 3. The paper reflects the authors' research and analysis truthfully and completely. 4. The paper properly credits the meaningful contributions of co-authors and co-researchers. 5. The results are appropriately placed in the context of prior and existing research. 6. All sources used are properly disclosed (correct citation). 7. All authors have been personally and actively involved in substantial work leading to the paper, and will take public responsibility for its content. We agree with the above statements and declare that this submission follows the policies of Swarm Intelligence as outlined in the Guide for Authors and in the Ethical Statement.

Data Availability Statement: Data sharing does not apply to this article as no datasets were generated or analyzed during the current study.

Acknowledgments: The authors would like to acknowledge Taif University Researchers Supporting Project number (TURSP-2020/292) Taif University, Taif, Saudi Arabia. The authors would like also to acknowledge Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R193), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Conflicts of Interest: The authors have no relevant financial or non-financial interest to disclose.

Abbreviations

The following abbreviations are used in this manuscript:

ACO	Ant Colony Optimization	GA	Genetic Algorithm
ALO	Ant Lion Optimizer	FOA	Glowworm Swarm Optimization
ABC	Artifical Bee Colony	GSA	Gravitational Search Algorithm
AFSA	Artificial Fish-Swarm Algorithm	GWO	Grey Wolf Optimizer
βHC	β-Hill Climbing	HSA	Harmony Search Algorithm
BA	Bat Algorithm	HSA	Harmony Search Algorithm
BBBC	Big Bang-Big Crunch	HGSO	Henry Gas Solubility Optimization
BMO	Bird Mating Optimizer	KHA	Krill Herd Algorithm
BOA	Butterfly Optimization Algorithm	LO	Lemurs Optimizer
CSOA	Cat Swarm Optimization Algorithm	MBO	Monarch Butterfly Optimization
CSS	Charged System Search	MVO	Multi-Verse Optimizer
CSO	Chicken Swarm Optimization	NFL	No Free Lunch
CHIO	Coronavirus Herd Immunity Optimizer	PSO	Particle Swarm Optimization
SCA	Cosine Algorithm	SOA	Seeker Optimization Algorithm
CSA	Crow Search Algorithm	SA	Simulated Annealing
CS	Cuckoo Search	SBA	Social-Based Algorithm
DE	Differential Evolution	TS	Tabu Search
EM	Electromagnetism-Like Mechanism	WCA	Water Cycle Algorithm
EO	Equilibrium Optimizer	WOA	Whale Optimization Algorithm
EA	Evolutionary Algorithm	WAC	Wisdom Of Artificial Crowds

References

1. Singh, S.P.; Dhiman, G.; Tiwari, P.; Jhaveri, R.H. A soft computing based multi-objective optimization approach for automatic prediction of software cost models. *Appl. Soft Comput.* **2021**, *113*, 107981. [CrossRef]
2. Varga, D. Full-Reference Image Quality Assessment Based on an Optimal Linear Combination of Quality Measures Selected by Simulated Annealing. *J. Imaging* **2022**, *8*, 224. [CrossRef] [PubMed]
3. Fong, S.; Zhuang, Y.; Tang, R.; Yang, X.S.; Deb, S. Selecting optimal feature set in high-dimensional data by swarm search. *J. Appl. Math.* **2013**, *2013*, 590614. [CrossRef]
4. Marichelvam, M.K.; Prabaharan, T.; Yang, X.S. A discrete firefly algorithm for the multi-objective hybrid flowshop scheduling problems. *IEEE Trans. Evol. Comput.* **2013**, *18*, 301–305. [CrossRef]
5. Rajamoorthy, R.; Arunachalam, G.; Kasinathan, P.; Devendiran, R.; Ahmadi, P.; Pandiyan, S.; Muthusamy, S.; Panchal, H.; Kazem, H.A.; Sharma, P. A novel intelligent transport system charging scheduling for electric vehicles using Grey Wolf Optimizer and Sail Fish Optimization algorithms. *Energy Sources Part A Recover. Util. Environ. Eff.* **2022**, *44*, 3555–3575. [CrossRef]
6. Rosso, S.; Uriati, F.; Grigolato, L.; Meneghello, R.; Concheri, G.; Savio, G. An optimization workflow in design for additive manufacturing. *Appl. Sci.* **2021**, *11*, 2572. [CrossRef]
7. Sharma, P.; Said, Z.; Kumar, A.; Nižetić, S.; Pandey, A.; Hoang, A.T.; Huang, Z.; Afzal, A.; Li, C.; Le, A.T.; et al. Recent advances in machine learning research for nanofluid-based heat transfer in renewable energy system. *Energy Fuels* **2022**, *36*, 6626–6658. [CrossRef]
8. Abasi, A.K.; Khader, A.T.; Al-Betar, M.A.; Naim, S.; Alyasseri, Z.A.A.; Makhadmeh, S.N. A novel hybrid multi-verse optimizer with K-means for text documents clustering. *Neural Comput. Appl.* **2020**, *32*, 17703–17729. [CrossRef]
9. Holland, J.H. Genetic algorithms and adaptation. In *Adaptive Control of Ill-Defined Systems*; Springer: Berlin/Heidelberg, Germany, 1984; pp. 317–333.
10. Koza, J.R.; Koza, J.R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*; MIT Press: Cambridge, MA, USA, 1992; Volume 1.
11. Back, T.; Hoffmeister, F.; Schwefel, H.P. A survey of evolution strategies. In *Proceedings of the Fourth International Conference on Genetic Algorithms*; Morgan Kaufmann Publishers: San Mateo, CA, USA, 1991; Volume 2.
12. Simon, D. Biogeography-based optimization. *IEEE Trans. Evol. Comput.* **2008**, *12*, 702–713. [CrossRef]
13. Deng, W.; Shang, S.; Cai, X.; Zhao, H.; Song, Y.; Xu, J. An improved differential evolution algorithm and its application in optimization problem. *Soft Comput.* **2021**, *25*, 5277–5298. [CrossRef]
14. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
15. Turabieh, H.; Al Azwari, S.; Rokaya, M.; Alosaimi, W.; Alharbi, A.; Alhakami, W.; Alnfiai, M. Enhanced harris hawks optimization as a feature selection for the prediction of student performance. *Computing* **2021**, *103*, 1417–1438. [CrossRef]
16. Abasi, A.K.; Khader, A.T.; Al-Betar, M.A.; Naim, S.; Alyasseri, Z.A.A.; Makhadmeh, S.N. An ensemble topic extraction approach based on optimization clusters using hybrid multi-verse optimizer for scientific publications. *J. Ambient Intell. Humaniz. Comput.* **2021**, *12*, 2765–2801. [CrossRef]
17. Abasi, A.K.; Khader, A.T.; Al-Betar, M.A.; Naim, S.; Makhadmeh, S.N.; Alyasseri, Z.A.A. A novel ensemble statistical topic extraction method for scientific publications based on optimization clustering. *Multimed. Tools Appl.* **2021**, *80*, 37–82. [CrossRef]
18. Goldberg, D.E.; Holland, J.H. Genetic Algorithms and Machine Learning 1988. Available online: https://dl.acm.org/doi/pdf/10.1145/168304.168305 (accessed on 28 September 2022).
19. Dorigo, M.; Di Caro, G. Ant colony optimization: A new meta-heuristic. In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Washington, DC, USA, 6–9 July 1999; Volume 2, pp. 1470–1477.
20. Karaboga, D. *An Idea Based on Honey Bee Swarm for Numerical Optimization*; Technical Report, Technical Report-tr06; Erciyes University: Talas/Kayseri, Turkey, 2005.
21. Geem, Z.W.; Kim, J.H.; Loganathan, G.V. A new heuristic optimization algorithm: Harmony search. *Simulation* **2001**, *76*, 60–68. [CrossRef]
22. Tan, Y.; Zhu, Y. Fireworks algorithm for optimization. In *International Conference in Swarm Intelligence*; Springer: Berlin/Heidelberg, 2010; pp. 355–364.
23. Lam, A.Y.; Li, V.O. Chemical-reaction-inspired metaheuristic for optimization. *IEEE Trans. Evol. Comput.* **2009**, *14*, 381–399. [CrossRef]
24. Al-Betar, M.A. β-Hill climbing: An exploratory local search. *Neural Comput. Appl.* **2017**, *28*, 153–168. [CrossRef]
25. Al-Betar, M.A.; Alyasseri, Z.A.A.; Awadallah, M.A.; Doush, I.A. Coronavirus herd immunity optimizer (CHIO). *Neural Comput. Appl.* **2021**, *33*, 5011–5042. [CrossRef]
26. He, S.; Wu, Q.; Saunders, J. A novel group search optimizer inspired by animal behavioural ecology. In Proceedings of the 2006 IEEE International Conference on Evolutionary Computation, Vancouver, BC, Canada, 16–21 July 2006; pp. 1272–1278.
27. Sadollah, A.; Bahreininejad, A.; Eskandar, H.; Hamdi, M. Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. *Appl. Soft Comput.* **2013**, *13*, 2592–2612. [CrossRef]
28. Dai, C.; Zhu, Y.; Chen, W. Seeker optimization algorithm. In *International Conference on Computational and Information Science*; Springer: Berlin/Heidelberg, 2006; pp. 167–176.
29. Ramezani, F.; Lotfi, S. Social-based algorithm (SBA). *Appl. Soft Comput.* **2013**, *13*, 2837–2856. [CrossRef]

30. Glover, F.; Laguna, M. Tabu search. In *Handbook of Combinatorial Optimization*; Springer: Berlin/Heidelberg, Germany, 1998; pp. 2093–2229.
31. Yampolskiy, R.V.; El-Barkouky, A. Wisdom of artificial crowds algorithm for solving NP-hard problems. *Int. J. Bio-Inspired Comput.* **2011**, *3*, 358–369. [CrossRef]
32. Erol, O.K.; Eksin, I. A new optimization method: Big bang–big crunch. *Adv. Eng. Softw.* **2006**, *37*, 106–111. [CrossRef]
33. Kaveh, A.; Talatahari, S. A novel heuristic optimization method: Charged system search. *Acta Mech.* **2010**, *213*, 267–289. [CrossRef]
34. Birbil, Ş.İ.; Fang, S.C. An electromagnetism-like mechanism for global optimization. *J. Glob. Optim.* **2003**, *25*, 263–282. [CrossRef]
35. Faramarzi, A.; Heidarinejad, M.; Stephens, B.; Mirjalili, S. Equilibrium optimizer: A novel optimization algorithm. *Knowl. Based Syst.* **2020**, *191*, 105190. [CrossRef]
36. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. GSA: A gravitational search algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [CrossRef]
37. 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]
38. 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]
39. 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]
40. Mirjalili, S. SCA: A sine cosine algorithm for solving optimization problems. *Knowl. Based Syst.* **2016**, *96*, 120–133. [CrossRef]
41. Mirjalili, S. The ant lion optimizer. *Adv. Eng. Softw.* **2015**, *83*, 80–98. [CrossRef]
42. Li, X.l. An optimizing method based on autonomous animats: Fish-swarm algorithm. *Syst. Eng. Theory Pract.* **2002**, *22*, 32–38.
43. Yang, X.S.; Gandomi, A.H. Bat algorithm: A novel approach for global engineering optimization. *Eng. Comput.* **2012**, *29*, 464–483. [CrossRef]
44. Askarzadeh, A. Bird mating optimizer: An optimization algorithm inspired by bird mating strategies. *Commun. Nonlinear Sci. Numer. Simul.* **2014**, *19*, 1213–1228. [CrossRef]
45. Arora, S.; Singh, S. Butterfly optimization algorithm: A novel approach for global optimization. *Soft Comput.* **2019**, *23*, 715–734. [CrossRef]
46. Chu, S.C.; Tsai, P.W.; Pan, J.S. Cat swarm optimization. In *Pacific Rim International Conference on Artificial Intelligence*; Springer: Berlin/Heidelberg, 2006; pp. 854–858.
47. Askarzadeh, A. A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. *Comput. Struct.* **2016**, *169*, 1–12. [CrossRef]
48. Yang, X.S.; Deb, S. Cuckoo search via Lévy flights. In Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), Coimbatore, India, 9–11 December 2009; pp. 210–214.
49. Meng, X.; Liu, Y.; Gao, X.; Zhang, H. A new bio-inspired algorithm: Chicken swarm optimization. In *International Conference in Swarm Intelligence*; Springer: Cham, Switzerland, 2014; pp. 86–94.
50. Mirjalili, S. Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput. Appl.* **2016**, *27*, 1053–1073. [CrossRef]
51. Deb, S.; Fong, S.; Tian, Z. Elephant search algorithm for optimization problems. In Proceedings of the 2015 Tenth International Conference on Digital Information Management (ICDIM), Jeju, Korea, 21–23 October 2015; pp. 249–255.
52. Yang, X.S. Firefly algorithm. *Nat. Inspired Metaheuristic Algorithms* **2008**, *20*, 79–90.
53. Yang, X.S. Flower pollination algorithm for global optimization. In *International Conference on Unconventional Computing and Natural Computation*; Springer: Berlin/Heidelberg, 2012; pp. 240–249.
54. 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]
55. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl. Based Syst.* **2015**, *89*, 228–249. [CrossRef]
56. Wang, G.G.; Deb, S.; Cui, Z. Monarch butterfly optimization. *Neural Comput. Appl.* **2019**, *31*, 1995–2014. [CrossRef]
57. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
58. Pan, W.T. A new fruit fly optimization algorithm: Taking the financial distress model as an example. *Knowl. Based Syst.* **2012**, *26*, 69–74. [CrossRef]
59. Krishnanand, K.; Ghose, D. Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions. *Swarm Intell.* **2009**, *3*, 87–124. [CrossRef]
60. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* **2019**, *97*, 849–872. [CrossRef]
61. Gandomi, A.H.; Alavi, A.H. Krill herd: A new bio-inspired optimization algorithm. *Commun. Nonlinear Sci. Numer. Simul.* **2012**, *17*, 4831–4845. [CrossRef]
62. Fathollahi-Fard, A.M.; Hajiaghaei-Keshteli, M.; Tavakkoli-Moghaddam, R. Red deer algorithm (RDA): A new nature-inspired meta-heuristic. *Soft Comput.* **2020**, *24*, 14637–14665. [CrossRef]
63. Trojovskỳ, P.; Dehghani, M. Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications. *Sensors* **2022**, *22*, 855. [CrossRef]

64. Oszust, M. Enhanced marine predators algorithm with local escaping operator for global optimization. *Knowl. Based Syst.* **2021**, *232*, 107467. [CrossRef]
65. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
66. Kappeler, P.M.; van Schaik, C.P. Evolution of primate social systems. *Int. J. Primatol.* **2002**, *23*, 707–740. [CrossRef]
67. Zimmermann, E.; Cepok, S.; Rakotoarison, N.; Zietemann, V.; Radespiel, U. Sympatric mouse lemurs in north-west Madagascar: A new rufous mouse lemur species (Microcebus ravelobensis). *Folia Primatol.* **1998**, *69*, 106–114. [CrossRef] [PubMed]
68. Powzyk, J.A.; Mowry, C.B. The feeding ecology and related adaptations of Indri indri. In *Lemurs*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 353–368.
69. LaTorre, A.; Muelas, S.; Peña, J.M. Benchmarking a hybrid DE-RHC algorithm on real world problems. In Proceedings of the 2011 IEEE Congress of Evolutionary Computation (CEC), New Orleans, LA, USA, 5–8 June 2011; pp. 1027–1033.
70. Wilcoxon, F. Individual comparisons by ranking methods. In *Breakthroughs in Statistics*; Springer: Berlin/Heidelberg, Germany, 1992; pp. 196–202.
71. Das, S.; Suganthan, P.N. *Problem Definitions and Evaluation Criteria for CEC 2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Problems*; Tehnical report; Jadavpur University, Nanyang Technological University: Kolkata, India, 2010; pp. 341–359.
72. Omran, M.G.H.; Clerc, M. APS 9: An improved adaptive population-based simplex method for real-world engineering optimization problems. *Appl. Intell.* **2018**, *48*, 1596–1608. [CrossRef]
73. Asafuddoula, M.; Ray, T.; Sarker, R. An adaptive differential evolution algorithm and its performance on real world optimization problems. In Proceedings of the 2011 IEEE Congress of Evolutionary Computation (CEC), New Orleans, LA, USA, 5–8 June 2011; pp. 1057–1062.
74. Korošec, P.; Šilc, J. The continuous differential ant-stigmergy algorithm applied to real-world optimization problems. In Proceedings of the 2011 IEEE Congress of Evolutionary Computation (CEC), New Orleans, LA, USA, 5–8 June 2011; pp. 1327–1334.
75. Zamuda, A.; Brest, J. On tenfold execution time in real world optimization problems with differential evolution in perspective of algorithm design. In Proceedings of the 2018 25th International Conference on Systems, Signals and Image Processing (IWSSIP), Maribor, Slovenia, 20–22 June 2018; pp. 1–5.
76. Elsayed, S.M.; Sarker, R.A.; Essam, D.L. GA with a new multi-parent crossover for solving IEEE-CEC2011 competition problems. In Proceedings of the 2011 IEEE Congress of Evolutionary Computation (CEC), New Orleans, LA, USA, 5–8 June 2011; pp. 1034–1040.
77. Reynoso-Meza, G.; Sanchis, J.; Blasco, X.; Herrero, J.M. Hybrid DE algorithm with adaptive crossover operator for solving real-world numerical optimization problems. In Proceedings of the 2011 IEEE Congress of Evolutionary Computation (CEC), New Orleans, LA, USA, 5–8 June 2011; pp. 1551–1556.
78. Singh, H.K.; Ray, T. Performance of a hybrid EA-DE-memetic algorithm on CEC 2011 real world optimization problems. In Proceedings of the 2011 IEEE Congress of Evolutionary Computation (CEC), New Orleans, LA, USA, 5–8 June 2011; pp. 1322–1326.
79. Omran, M.G.; Alsharhan, S.; Clerc, M. A modified Intellects-Masses Optimizer for solving real-world optimization problems. *Swarm Evol. Comput.* **2018**, *41*, 159–166. [CrossRef]
80. Gothania, B.; Mathur, G.; Yadav, R. Accelerated Artificial Bee Colony Algorithm for Parameter Estimation of Frequency-modulated Sound Waves. *Int. J. Electron. Commun. Eng.* **2014**, *7*, 63–74.
81. Wang, H.; Yi, J.H. An improved optimization method based on krill herd and artificial bee colony with information exchange. *Memetic Comput.* **2018**, *10*, 177–198. [CrossRef]
82. Han, X.; Zhao, L.; Wen, J.; Ai, X.; Liu, J.; Yang, D. Transmission network expansion planning considering the generators' contribution to uncertainty accommodation. *CSEE J. Power Energy Syst.* **2017**, *3*, 450–460. [CrossRef]
83. Abasi, A.K.; Khader, A.T.; Al-Betar, M.A.; Naim, S.; Makhadmeh, S.N.; Alyasseri, Z.A.A. Link-based multi-verse optimizer for text documents clustering. *Appl. Soft Comput.* **2020**, *87*, 106002. [CrossRef]
84. Abasi, A.K.; Khader, A.T.; Al-Betar, M.A.; Alyasseri, Z.A.A.; Makhadmeh, S.N.; Al-laham, M.; Naim, S. A Hybrid Salp Swarm Algorithm with β-Hill Climbing Algorithm for Text Documents Clustering. In *Evolutionary Data Clustering: Algorithms and Applications*; Springer: Singapore, 2021; pp. 129–161.
85. Alyasseri, Z.A.A.; Khader, A.T.; Al-Betar, M.A.; Abasi, A.K.; Makhadmeh, S.N. EEG signals denoising using optimal wavelet transform hybridized with efficient metaheuristic methods. *IEEE Access* **2019**, *8*, 10584–10605. [CrossRef]
86. Alyasseri, Z.A.A.; Khadeer, A.T.; Al-Betar, M.A.; Abasi, A.; Makhadmeh, S.; Ali, N.S. The effects of EEG feature extraction using multi-wavelet decomposition for mental tasks classification. In Proceedings of the International Conference on Information and Communication Technology (ICICT '19), Baghdad, Iraq, 15–16 April 2019; pp. 139–146.
87. Alyasseri, Z.A.A.; Abasi, A.K.; Al-Betar, M.A.; Makhadmeh, S.N.; Papa, J.P.; Abdullah, S.; Khader, A.T. EEG-Based Person Identification Using Multi-Verse Optimizer as Unsupervised Clustering Techniques. In *Evolutionary Data Clustering: Algorithms and Applications*; Springer: Singapore, 2021; pp. 89–110.
88. Wang, Y.; Li, X.; Wang, J. A neurodynamic optimization approach to supervised feature selection via fractional programming. *Neural Netw.* **2021**, *136*, 194–206. [CrossRef]

89. Abasi, A.K.; Khader, A.T.; Al-Betar, M.A.; Naim, S.; Makhadmeh, S.N.; Alyasseri, Z.A.A. A text feature selection technique based on binary multi-verse optimizer for text clustering. In Proceedings of the 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, 9–11 April 2019; pp. 1–6.
90. Abasi, A.K.; Khader, A.T.; Al-Betar, M.A.; Naim, S.; Makhadmeh, S.N.; Alyasseri, Z.A.A. An improved text feature selection for clustering using binary grey wolf optimizer. In *Proceedings of the 11th National Technical Seminar on Unmanned System Technology 2019*; Springer: Singapore, 2021; pp. 503–516.
91. Alomari, O.A.; Makhadmeh, S.N.; Al-Betar, M.A.; Alyasseri, Z.A.A.; Doush, I.A.; Abasi, A.K.; Awadallah, M.A.; Zitar, R.A. Gene selection for microarray data classification based on Gray Wolf Optimizer enhanced with TRIZ-inspired operators. *Knowl. Based Syst.* **2021**, *223*, 107034. [CrossRef]
92. Makhadmeh, S.N.; Khader, A.T.; Al-Betar, M.A.; Naim, S.; Abasi, A.K.; Alyasseri, Z.A.A. Optimization methods for power scheduling problems in smart home: Survey. *Renew. Sustain. Energy Rev.* **2019**, *115*, 109362. [CrossRef]
93. Makhadmeh, S.N.; Khader, A.T.; Al-Betar, M.A.; Naim, S.; Alyasseri, Z.A.A.; Abasi, A.K. Particle swarm optimization algorithm for power scheduling problem using smart battery. In Proceedings of the 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, 9–11 April 2019; pp. 672–677.
94. Makhadmeh, S.N.; Al-Betar, M.A.; Alyasseri, Z.A.A.; Abasi, A.K.; Khader, A.T.; Damaševičius, R.; Mohammed, M.A.; Abdulkareem, K.H. Smart Home Battery for the Multi-Objective Power Scheduling Problem in a Smart Home Using Grey Wolf Optimizer. *Electronics* **2021**, *10*, 447. [CrossRef]
95. Makhadmeh, S.N.; Khader, A.T.; Al-Betar, M.A.; Naim, S.; Abasi, A.K.; Alyasseri, Z.A.A. A novel hybrid grey wolf optimizer with min-conflict algorithm for power scheduling problem in a smart home. *Swarm Evol. Comput.* **2021**, *60*, 100793. [CrossRef]