

Article

Snow Leopard Optimization Algorithm: A New Nature-Based Optimization Algorithm for Solving Optimization Problems

Petr Coufal ¹, Štěpán Hubálovský ^{1,*}, Marie Hubálovská ² and Zoltan Balogh ³

¹ Faculty of Science, University of Hradec Kralove, Rokitanskeho 62, 500 03 Hradec Kralove, Czech Republic; petr.coufal@uhk.cz

² Faculty of Education, University of Hradec Kralove, Rokitanskeho 62, 500 03 Hradec Kralove, Czech Republic; marie.hubalovska@uhk.cz

³ Faculty of Natural Sciences, The Constantine the Philosopher University in Nitra, A. Hlinku 1, 949 74 Nitra, Slovakia; zbalogh@ukf.sk

* Correspondence: stepan.hubalovsky@uhk.cz

Abstract: Numerous optimization problems have been defined in different disciplines of science that must be optimized using effective techniques. Optimization algorithms are an effective and widely used method of solving optimization problems that are able to provide suitable solutions for optimization problems. In this paper, a new nature-based optimization algorithm called Snow Leopard Optimization Algorithm (SLOA) is designed that mimics the natural behaviors of snow leopards. SLOA is simulated in four phases including travel routes, hunting, reproduction, and mortality. The different phases of the proposed algorithm are described and then the mathematical modeling of the SLOA is presented in order to implement it on different optimization problems. A standard set of objective functions, including twenty-three functions, is used to evaluate the ability of the proposed algorithm to optimize and provide appropriate solutions for optimization problems. Also, the optimization results obtained from the proposed SLOA are compared with eight other well-known optimization algorithms. The optimization results show that the proposed SLOA has a high ability to solve various optimization problems. Also, the analysis and comparison of the optimization results obtained from the SLOA with the other eight algorithms shows that the SLOA is able to provide more appropriate quasi-optimal solutions and closer to the global optimal, and with better performance, it is much more competitive than similar algorithms.

Keywords: optimization; optimization algorithm; nature-based algorithm; snow leopard; optimization problem



Citation: Coufal, P.; Hubálovský, Š.; Hubálovská, M.; Balogh, Z. Snow Leopard Optimization Algorithm: A New Nature-Based Optimization Algorithm for Solving Optimization Problems. *Mathematics* **2021**, *9*, 2832. <https://doi.org/10.3390/math9212832>

Academic Editors: Árpád Bűrmen and Tadej Tuma

Received: 16 October 2021

Accepted: 3 November 2021

Published: 8 November 2021

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



Copyright: © 2021 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

1.1. Motivation

The goal of optimization is to find the best acceptable answer, given the limitations and needs of the problem [1]. For a problem, there may be different solutions, and to compare them and select the optimal solution, a function called the objective function is defined. The choice of this function depends on the nature of the problem. However, choosing the suitable objective function is one of the most important optimization steps [2]. An optimization problem can be defined from a mathematical point of view using the three main parts of variables, objective functions, and constraints [3]. Once the optimization problem is mathematically modelled, it must be optimized using the appropriate method.

Optimization algorithms are among the optimization problem solving methods that are able to provide suitable solutions for optimization problems based on random scanning of the search space without the need for gradient information. In recent years, many optimization algorithms have been designed by scientists to solve optimization problems. These algorithms are based on simulations of various natural phenomena, the laws of physics, the biological sciences, the behavior of animals, insects, and living things in nature.

The main question that arises in the study of optimization algorithms is that according to the algorithms that have been introduced in recent years, is there a need to develop and design new algorithms? The answer to this question lies in the No Free Lunch (NFL) theorem [4]. According to the NFL theorem, a particular optimization algorithm will not be able to solve all optimization problems simply because it has a powerful performance in solving several optimization problems. In fact, an optimization algorithm may be the best optimizer in solving one optimization problem, but it fails in solving another problem. This is because each optimization problem has its own complexity and nature. The NFL theorem prompts researchers to design new optimization algorithms that can solve optimization problems in different domains and applications. NFL theorem also motivated the authors of this paper to design and introduce a new optimization algorithm for solving optimization problems.

1.2. Research Gap

Although many optimization algorithms have been introduced to solve the optimization problems, achieving more suitable quasi-optimal solutions closer to the global optimal is still a major challenge in solving and optimizing various optimization problems. The key issue in the studies of optimization algorithms is that since optimization algorithms are considered stochastic-based methods, they do not guarantee provided solutions as the global optimal. Therefore, it is always possible to develop new optimization algorithms that are able to provide quasi-optimal solutions to the global optimal. In order to solve more accurate and more effective optimization problems in different sciences, it is necessary to design newer algorithms with higher abilities in achieving quasi-optimal solutions closer to global optimal solutions. Therefore, in this study, it has been attempted to design a new optimization algorithm that can provide more suitable solutions for optimization issues.

One of the disadvantages of most optimization algorithms is that the process of updating their population depends too much on the best member of the population. This can lead to early convergence of the algorithm or entrapment in local optimal areas. The reason for the lack of optimal convergence in these methods is that it does not allow members of the population to properly scan the search space in different directions. The advantage of the proposed SLOA is that it does not rely on the best member of the population and increases the search power of the algorithm. In SLOA, the algorithm population is updated in four different phases. None of these four phases relies on the best member of the population. In the first phase, simulating the zigzag motion of snow leopards is very effective in accurately scanning the search space and exiting local optimal solutions. In the second phase, the behavior of snow leopards during hunting, which is at two different speeds towards the prey, increases the power of exploitation and the convergence of the algorithm towards the solution. In the third phase, the reproduction process creates random solutions in different areas of the search space, which increases the exploration power of the algorithm. In the fourth phase, mortality gives the algorithm the advantage of helping the algorithm evolve by eliminating weak population members and preventing the search for non-optimal areas.

1.3. Contribution

The innovation and contribution of this paper is in designing a new optimization algorithm called Snow Leopard Optimization Algorithm (SLOA) in order to solve optimization problems more effectively in different sciences. The novelty of the proposed method is to simulate the behavior and social life of snow leopards with a focus on travel routes, hunting, reproduction, and mortality. In the algorithms that have been introduced so far, researchers have not used the optimization process in the social life of snow leopards. The contributions proposed by this study are as follows:

- (i) The main idea in designing SLOA is simulation of the different behaviors of snow leopards that are inspired by nature. In SLOA, the natural behaviors of snow leopards are modeled in four phases including travel routes, hunting, reproduction, and mortality.

- (ii) The proposed SLOA is mathematically modeled and simulated for use in solving and optimizing various problems.
- (iii) Twenty-three sets of standard objective functions of different types are employed to evaluate the power of the proposed SLOA in providing quasi-optimal and effective solutions.
- (iv) In order to further analyze the SLOA and evaluate the quality of the obtained optimization results, the performance of the SLOA is compared with eight well-known optimization algorithms.

Optimization algorithms are applied in all disciplines of science and real-world problems where the optimization process or problem is defined and designed. The proposed SLOA can be applied to minimize or maximize various objective functions. SLOA can be applied in optimal designs and engineering sciences where decision variables must be well selected to optimize device performance. In data mining, medical science, clustering, and in general in any application that faces optimization, the proposed SLOA can be applied.

1.4. Paper Organization

The rest of the paper is organized in such a way that lecture review is presented in Section 2. The proposed SLOA is introduced in Section 3. Simulation studies on the performance of the SLOA are presented in Section 4. The performance of the SLOA from the perspective of exploration and exploitation is analyzed and discussed in Section 5. Finally, conclusions and several suggestions for future studies are presented in Section 6.

2. Lecture Review

Optimization problem solving methods include two categories of deterministic methods and stochastic methods [5].

The use of basic optimization methods including linear programming, integer programming, dynamic programming, and nonlinear programming is associated with disadvantages. The most important disadvantage of these methods is being time consuming in solving big problems using them. In a way, even with today's advanced computing technologies, solving a large-scale problem with the mentioned techniques takes several years [6]. The emergence of these problems has forced researchers to adjust their expectations to find the best possible answer and to be satisfied with good enough answers, that even for large-scale problems, appropriate solutions can be reached in a reasonable time [7].

The weakness of methods such as gradients and numerical calculations in solving optimization problems has led to the creation of a special type of intelligent search algorithms called population-based optimization algorithms. Population-based optimization algorithms are a kind of stochastic methods that are inspired by nature and its mechanisms [8]. These methods try to send their initial population to the global optimal and provide appropriate solutions close to the global optimal in a reasonable time [9].

The basis of the performance of optimization algorithms is that they first produce a number of solutions which is the population of the algorithm, then in an iterative-based process without the use of derivative information and only based on the simulation of the collective intelligence of the algorithm members improves these initial solutions [10].

Every optimization problem has a precise and basic solution called global optimal [11]. Given that optimization algorithms are stochastic methods and improve the initial proposed solutions during successive iterations, it is possible that the optimization algorithms will not be able to provide exactly the global optimal. For this reason, the solutions obtained using optimization algorithms for an optimization problem are called quasi-optimal [12]. In evaluating several quasi-optimal solutions to an optimization problem, the most appropriate solution is closer to the global optimal. The main reason for the design of numerous optimization algorithms by researchers is to provide quasi-optimal solutions that are more appropriate and closer to the global optimal solution. In this regard, optimization algorithms have been applied to solve optimization problems in different branches of science such as microwave for design of rectangular microstrip antennas [13], electromagnetic prob-

lems [14], Proportional-Integral-Derivative (PID) control [15], Flexible Job Planning (FJSP) problem [16], force analysis and optimization of kinematic parameters [17], simultaneous optimization of distributed generation [18], optimization of complex system reliability [19], design of model-based fuzzy controllers for networked control systems [20], design of the stable robot controller [21], and the Maki-Thompson rumor model [22].

Optimization algorithms have developed over the years based on various natural, physical, game, genetic, and any theory or process that has an evolving nature.

Genetic Algorithm (GA) is one of the famous, oldest, and widely used optimization methods that is inspired by the science of genetics and Darwin's theory of evolution, and is based on the survival of the fittest or natural selection. GA begins with the production of a population of chromosomes (the initial population of chromosomes in GA is randomly generated and bound to the top and bottom of the problem variables). In the next step, the generated data structures (chromosomes) are evaluated. Chromosomes that provide better values for the objective function of the problem have a higher chance to be selected as parents for reproduction than weaker solutions [23]. Although GA has simple concepts and can be easily implemented, having several control parameters and being time consuming is the most important disadvantage of GA.

Particle Swarm Optimization (PSO) is another popular and widely used algorithm that is inspired by the collective behavior of birds or fish in nature. In PSO, a group of birds or fish are looking for food in an environment where there is only one piece of food. None of the birds know the location of the food and only know the distance to the food. One of the best strategies is to follow a bird that is closer to the food. In other words, every bird or fish, in addition to its own experience, also trusts the bird or fish that is closest to the food [24]. The main disadvantages of PSO are that it is easy to fall into local optimal in high-dimensional problems, and it has a low convergence rate in the iterative process.

Gravitational Search Algorithm (GSA) is a physics-based algorithm that is introduced based on simulation of gravitational force and Newton's laws of motion. According to the theory of gravity, objects that are at different distances from each other exert a gravitational force on each other. In GSA, the mass of objects is determined based on the values of the objective function. Objects that are in a better position in the search space pull other objects towards themselves based on simulations of gravity force and the laws of motion [25]. Slow convergence, tendency to become trapped in local optimal, and having control parameters are the main disadvantages of GSA.

Teaching-Learning Based Optimization (TLBO) is a population-based optimization method that is designed based on simulation of behaviors of the teacher and students in a classroom. In TLBO, the best member of the population is considered the teacher and the rest of the population is considered the students of the class. TLBO has two phases called teaching phase and learning phase. In the teaching phase, the teacher teaches her/his knowledge to the students and in the learning phase, the students share their information with each other [26]. The main disadvantages of TLBO are that consumes lot of memory space and it involves lot of iterations, so TLBO is a time-consuming algorithm.

Gray Wolf Optimizer (GWO) is a nature-based optimizer that is introduced based on simulation of behaviors of grey wolves in nature. GWO mimics the hierarchical leadership and hunting mechanism of the gray wolves. Four types of gray wolves named alpha, beta, delta, and omega are used to simulate hierarchical leadership. In GWO, alpha is the best member of the population, beta and delta are the second and third best members of the population, and the rest of the wolves are omega. In addition, three stages including search for prey, encircling prey, and attacking prey are used to simulate hunting mechanism [27], low solving accuracy, slow convergence rate, and bad local searching ability are several disadvantages of GWO.

Grasshopper Optimization Algorithm (GOA) is a swarm-based algorithm that is introduced based on the simulation of behavior of grasshopper swarms in nature. In GOA, the group movement of grasshoppers towards food sources is imitated and simulated. A mathematical model is proposed for simulation of attraction and repulsion forces between

the grasshoppers. Attraction forces encouraged grasshoppers to exploit promising regions, whereas repulsion forces let them explore the search space [28]. The most important disadvantages of the GOA are slow convergence speed, the fact that it's time-consuming, and having control parameters.

Marine Predators Algorithm (MPA) is a nature-based algorithm that is inspired by the movement strategies that marine predators use when trapping their prey in the oceans. The predominant search behavior and strategy of marine predators for hunting is modeled using the Levy flight method. MPA has three phases due to the different speeds of the predators and the prey: Phase 1: When the prey moves faster than the predator, Phase 2: When the prey and the predator move at almost the same speed, and Phase 3: When the predator is moving faster than the prey [29]. One of the main disadvantages of MPA is that it requires huge number of iterations, especially for nonlinear optimization problems.

Tunicate Swarm Algorithm (TSA) is a bio-inspired algorithm that is introduced based on imitation of swarm behaviors of tunicates and jet propulsion during the navigation and foraging process. In TSA, two behaviors of tunicates including jet propulsion and swarm intelligence are employed for finding the food sources. In order to model the jet propulsion behavior, tunicates must comply with three conditions including remaining close to the best search agent, moving towards the position of best search agent, and avoiding the conflicts between search agents. In order to model the swarm intelligence behavior, positions of search agents should be updated based on the best optimal solution [30]. Poor convergence in solving high-dimensional multimodal problems, having control parameters, and complex calculations are the main disadvantages of the TSA.

3. Snow Leopard Optimization Algorithm (SLOA)

In this section, the snow leopard is introduced first. Then, based on simulating the habits and natural behaviors of snow leopards, a new optimization algorithm called Snow Leopard Optimization Algorithm (SLOA) is developed. Mathematical modeling and formulation of the proposed SLOA for implementation in solving optimization problems is presented.

3.1. Snow Leopard

The snow leopard (*Panthera uncia*) is a species of *Panthera* native that lives in the high mountains of South and Central Asia. Snow leopards live in mountainous and alpine areas at altitudes of 3000 to 4500 m from eastern Afghanistan, the Himalayas, and the Tibetan Plateau to southern Siberia, Mongolia, and western China [31].

The fur of snow leopards is whitish to gray with black spots on neck and head, with larger rosettes on the back, flanks, and bushy tail. The snow leopard's belly is whitish. Its eyes are grey or pale green. Its nasal cavities are large. Its forehead is domed and its muzzle is short. The fur is thick with hairs between 5 and 12 cm long. Its body is stocky, short-legged, and slightly smaller than the other cats of the genus *Panthera*, reaching a shoulder height of 56 cm, and ranging in head to body size from 75 to 150 cm. Its tail is 80 to 105 cm long [32]. It weighs between 22 and 55 kg, with an occasional large male reaching 75 kg, and small female of under 25 kg. Its canine teeth are 28.6 mm long and are more slender than those of the other *Panthera* species [33].

Snow leopards have different behaviors and habits, including how they move towards each other and travel routes, how they hunt, reproduce, and mortality. The modeling of these natural behaviors has been used in the design of the proposed SLOA. In this design, modeling of four optimal natural behaviors in the life of snow leopards is used.

The first behavior is travel routes and movement. Modeling the zig-zag pattern movements of snow leopards as they move and follow each other leads to a more efficient search of the search space and crossing the optimal local areas.

The second behavior is how to hunt. Modeling the movements of snow leopards in order to hunt prey leads to the convergence of the optimization algorithm towards the optimal areas.

The third behavior is reproduction. The reproduction of snow leopards can be modeled as a combination of two members of the population, which leads to the production of a new member that may improve the performance of the algorithm in achieving optimal areas.

The fourth behavior is mortality. Modeling the mortality of weak snow leopards leads to the elimination of solutions and inappropriate members of the algorithm. This will remove members in inappropriate areas from the search space. In addition, modeling this behavior leads to the algorithm population remaining constant during the algorithm iterations.

3.2. Mathematical Modeling

In the proposed SLOA, each snow leopard is a member of the algorithm population. A certain number of snow leopards as search agents are members of the SLOA. In population-based optimization algorithms, population members are identified using a matrix called the population matrix. The number of rows in the population matrix is equal to the number of members in the population, and the number of columns in this matrix is equal to the number of variables in the optimization problem. The population matrix is specified as a matrix representation using Equation (1).

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N \times m} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,d} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots & & \vdots \\ x_{i,1} & \cdots & x_{i,d} & \cdots & x_{i,m} \\ \vdots & & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,d} & \cdots & x_{N,m} \end{bmatrix}_{N \times m} \tag{1}$$

where, X is the population of snow leopard, X_i is the i th snow leopard, $x_{i,d}$ is the value for d th problem variable suggested by i th snow leopard, N is the number of snow leopard in algorithm population, and m is the number of problem variables.

The position of each snow leopard as a member of the population in the problem-solving space determines the values for the problem variables. Therefore, for each snow leopard, a value can be calculated for the objective function of the problem. The values of the objective function are specified by a vector using Equation (2).

$$F = \begin{bmatrix} f_1 \\ \vdots \\ f_i \\ \vdots \\ f_N \end{bmatrix}_{N \times 1} = \begin{bmatrix} f(X_1) \\ \vdots \\ f(X_i) \\ \vdots \\ f(X_N) \end{bmatrix}_{N \times 1} \tag{2}$$

here, F is the vector of objective function and F_i is the value for objective function of problem obtained based on i th snow leopard.

Members of the population are updated in the proposed SLOA based on simulating the natural behaviors of snow leopards in four phases: displacement, hunting, reproduction, and mortality. The mathematical modeling of these four phases and the mentioned natural behaviors are presented in the following subsections.

3.2.1. Phase 1: Travel Routes and Movement

Snow leopards, like other cats, use scent signs to show their locations and travel routes. These signs are usually caused by scraping the ground with the hind feet before depositing urine or scat [34]. Snow leopards also move in a zig-zag pattern in indirect lines [35]. So, snow leopards can move towards or follow each other based on this natural behavior.

This phase of the proposed SLOA is mathematically modeled using Equations (3)–(5).

$$x_{i,d}^{P1} = x_{i,d} + r \times (x_{k,d} - I \times x_{i,d}) \times \text{sign}(F_i - F_k), \tag{3}$$

$$k \in 1, 2, 3, \dots, N, \quad d = 1, 2, 3, \dots, m$$

$$X_i = \begin{cases} X_i^{P1}, & F_i^{P1} < F_i \\ X_i, & \text{else} \end{cases} \tag{4}$$

$$I = \text{round}(1 + r) \tag{5}$$

here, $x_{i,d}^{P1}$ is the new value for d th problem variable obtained by i th snow leopard based on phase 1, r is a random number in interval of $[0, 1]$, k is the row number of selected snow leopard for guiding i th snow leopard in d th axis, X_i^{P1} is the updated location of i th snow leopard based on phase 1, and F_i^{P1} is its objective function value.

3.2.2. Phase 2: Hunting

In the second phase of updating the members of the population, the behavior of snow leopards during hunting and attacking prey is used. The process and method of hunting a snow leopard, based on an observation recorded in Hemis National Park, is that the snow leopard uses rocky cliffs to cover itself when approaching its prey. After reaching a distance of 40 m from the prey, the snow leopard first walked slowly for the first 15 m, then ran the last 25 m, and finally killed the prey by biting its neck [36].

The natural behavior of snow leopards during hunting is mathematically modeled using Equations (6)–(8). Equation (6) specifies the location of the prey for i th snow leopard. Equation (7) simulates how a snow leopard moves toward its prey. According to observations, a snow leopard walks about 0.375% of the distance to the prey and then runs 0.625% of the distance to the prey. Therefore, a parameter called P is used to simulate this type of motion in Equation (7). Parameter P shows the percentage from the distance to the prey that the snow leopard walks. In the simulation of the proposed SLOA, the value of this parameter based on observations is considered 0.375. In Equation (8), the new position of the snow leopard after the attack on the prey is simulated. For this purpose, an effective update is used in which the new position is acceptable to an algorithm member if the value of the objective function in the new position is more appropriate than the previous position.

$$p_{i,d} = x_{j,d}, \quad d = 1, 2, 3, \dots, m \tag{6}$$

$$x_{i,d}^{P2} = x_{i,d} + r \times ((p_{i,d} - x_{i,d}) \times P + (p_{i,d} - 2 \times x_{i,d}) \times (1 - P)) \times \text{sign}(F_i - F_p), \tag{7}$$

$$X_i = \begin{cases} X_i^{P2}, & F_i^{P2} < F_i \\ X_i, & \text{else} \end{cases} \tag{8}$$

here, $p_{i,d}$ is the d th dimension of location of prey considered for i th snow leopard, F_p is the objective function value based on location of prey, $x_{i,d}^{P2}$ is the new value for d th problem variable obtained by i th snow leopard based on phase 2, and F_i^{P2} is its objective function value.

3.2.3. Phase 3: Reproduction

In this phase, based on the natural reproductive behavior of snow leopards, new members equal to half the total population are added to the population of the algorithm. In fact, it is assumed that a cub will be born based on the mating of both snow leopards. The reproduction process of snow leopards is mathematically modeled based on the mentioned concepts using Equation (9).

$$C_l = \frac{x_l + X_{N-l+1}}{2}, \quad l = 1, 2, 3, \dots, \frac{N}{2} \tag{9}$$

here, C_l is the l th cub which is born from the mating of two snow leopards.

3.2.4. Phase 4: Mortality

Living things are always in danger of dying. Although reproduction increases the population of snow leopards, the number of snow leopards remains constant during the replication of the algorithm due to mortality and losses. In the proposed SLOA, it is assumed that in each replication after reproduction, snow leopards face mortality exactly as the number of puppies born. The criterion of snow leopard mortality in the SLOA is the value of the objective function. Therefore, snow leopards that have a weaker objective function are more prone to death. Also, some born cubs may die due to having poor objective function.

3.3. Flowchart of SLOA

In the proposed SLOA, snow leopards are updated in each iteration according to the first and second phases, then the population of the algorithm according to the third and fourth phases is faced with natural processes of reproduction and mortality.

These steps of the SLOA are repeated until the stop condition is reached. After fully implementing the SLOA on an optimization problem, the SLOA makes available the best obtained solution as the best quasi-optimal solution. The various stages of implementation of the SLOA are specified as flowcharts in Figure 1 and its pseudocode is presented in Algorithm 1.

Algorithm 1 Pseudocode of SLOA

Start SLOA.

1. Input problem information: variables, objective function, and constraints.
 2. Set number of snow leopard (N) and iterations (T).
 3. Generate an initial population matrix at random.
 4. Evaluate the objective function.
 5. **For** $t = 1:T$
 6. **Phase 1:** travel routes and movement
 7. **For** $i = 1:N$
 8. **For** $d = 1:m$
 9. Calculate $x_{i,d}^{P1}$ using Equation (3) and Equation (5).
 10. **End**
 11. Update X_i using Equation (4)
 12. **End**
 13. **Phase 2:** hunting
 14. **For** $i = 1:N$
 15. **For** $d = 1:m$
 16. Calculate location of prey $p_{i,d}$ using Equation (6)
 17. Calculate $x_{i,d}^{P2}$ using Equation (7).
 18. **End**
 19. Update X_i using Equation (8).
 20. **End**
 21. **Phase 3:** reproduction
 22. **For** $l = 1:0.5 \times N$
 23. Generate cub C_l using Equation (9)
 24. **End**
 25. **Phase 4:** mortality
 26. Adjust the number of snow leopards to N due to mortality based on criterion of the objective function.
 27. Save best quasi-optimal solution obtained with the SLOA so far.
 28. **End For** $t = 1:T$
 29. Output best quasi-optimal solution obtained with the SLOA.
- End SLOA.**
-

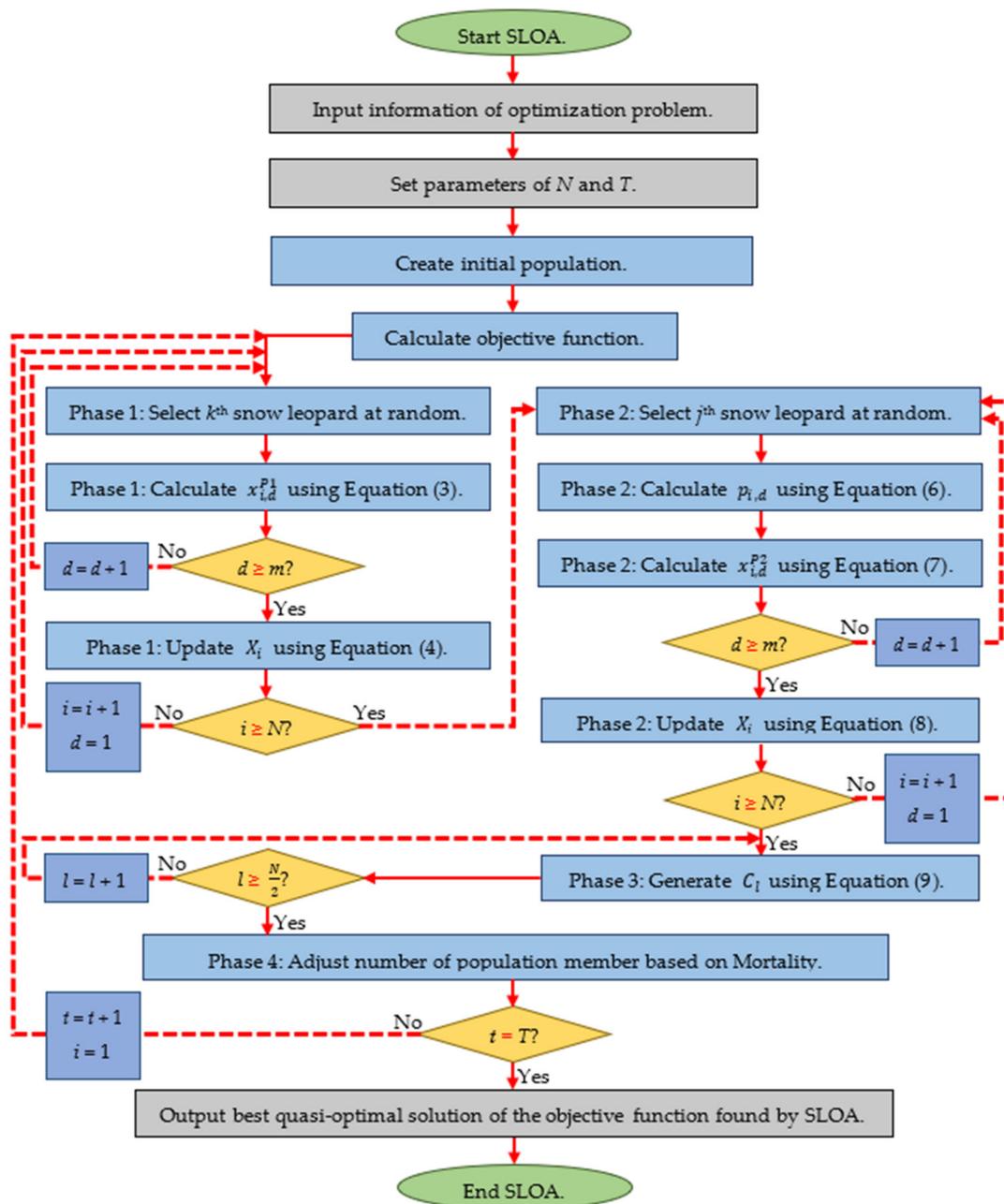


Figure 1. Flowchart of SLOA.

4. Simulation Studies and Results

In this section, the performance of the proposed SLOA in optimization and providing effective solutions to optimization problems are studied. For this purpose, a standard set of twenty-three objective functions of three different types of unimodal, high-dimensional multimodal, and fixed-dimensional multimodal has been used. The complete information of these objective functions is specified in Appendix A and in Tables A1–A3. The optimization results obtained using the SLOA are compared with the performance of eight optimization algorithms including Genetic Algorithm (GA) [23], Particle Swarm Optimization (PSO) [24], Gravitational Search Algorithm (GSA) [25], Teaching-Learning Based Optimization (TLBO) [26], Gray Wolf Optimizer (GWO) [27], Grasshopper Optimization Algorithm (GOA) [28], Marine Predators Algorithm (MPA) [29], and Tunicate Swarm Algorithm (TSA) [30]. The proposed SLOA, as well as eight compared algorithms, are implemented in twenty independent implementations on the optimization of twenty-three

objective functions F1 to F23. The most important criterion for determining the superiority of optimization algorithms is the value of the objective function. Although the algorithm convergence speed is an important criterion in the performance of optimization algorithms, the main purpose of optimizing an optimization problem is to provide a suitable quasi-optimal solution. An algorithm may have a high convergence speed but converge to local or unsuitable solutions. In fact, in optimizing an objective function, it is a superior algorithm that can offer a better solution and with a more optimal objective function value. For this reason, the results of implementation and performance of optimization algorithms on the set of the mentioned objective functions have been reported using the two indicators of the average of the best obtained solutions from twenty independent executions for the objective function (ave) and the standard deviation of these best obtained solutions (std). These two indicators can be calculated using Equations (10) and (11). The values used for the main controlling parameters of the comparative optimization algorithms are specified in Table 1.

$$ave = \frac{1}{20} \times \sum_{i=1}^{20} BOS_i, \tag{10}$$

$$std = \sqrt{\frac{1}{20} \times \sum_{i=1}^{20} (BOS_i - ave)^2}, \tag{11}$$

where, BOS_i is the best obtained solution in i th independent implementation.

Table 1. Parameter values for the comparative algorithms.

Algorithm	Parameter	Value
GA	Type	Real coded
	Selection	Roulette wheel (Proportionate)
	Crossover	Whole arithmetic (Probability = 0.8, $\alpha = [-0.5, 1.5]$)
	Mutation	Gaussian (Probability = 0.05)
PSO	Topology	Fully connected
	Cognitive and social constant	$C_1 = 2, C_2 = 2$
	Inertia weight	Linear reduction from 0.9 to 0.1
	Velocity limit	10% of dimension range
GSA	Alpha, G_0, R_{norm}, R_{power}	20, 100, 2, 1
TLBO	T_F : teaching factor random number	$T_F = \text{round} [(1 + rand)]$ $rand$ is a random number between [0–1].
GWO	Convergence parameter (a)	a : Linear reduction from 2 to 0.
GOA	C_{min}, C_{max}, l, f r_1, r_2, r_3 are random numbers in [0–1].	0.0001, 1, 1.5, 0.5
TSA	P_{min} and P_{max} c_1, c_2, c_3	1, 4 random numbers lie in the range of [0–1].
MPA	Constant number	$p = 0.5$
	Random vector	R is a vector of uniform random numbers in [0–1].
	Fish Aggregating Devices (FADs)	$FADs = 0.2$
	Binary vector	$U = 0$ or 1

4.1. Evaluation of Unimodal Objective Functions

Seven F1 to F7 functions of the unimodal type are selected to analyze the ability of optimization algorithms to provide quasi-optimal solutions for the unimodal objective functions. The results of optimization of these objective functions using the proposed SLOA as well as eight other algorithms are presented in Table 2. The optimization results show that the proposed SLOA has been able to converge to the global optimal solution in solving F1 and F6 functions. SLOA is the first best optimizer in optimizing the F2, F3, F4, F5, and F7 functions. The simulation results show that in optimizing the F1, F2, F3, and F4 functions, the performance of the proposed SLOA is significantly superior to the eight compared algorithms.

Table 2. Optimization results of SLOA and other algorithms on unimodal test function.

		GA	PSO	GSA	TLBO	GWO	GOA	TSA	MPA	SLOA
F1	Ave	13.2405	1.77×10^{-5}	2.03×10^{-17}	8.34×10^{-60}	1.09×10^{-58}	8.62×10^{-8}	7.71×10^{-38}	3.27×10^{-21}	0
	std	4.77×10^{-15}	6.44×10^{-21}	1.14×10^{-32}	4.94×10^{-76}	5.14×10^{-74}	5.32×10^{-22}	7.00×10^{-21}	4.62×10^{-21}	0
F2	Ave	2.4794	0.3411	2.37×10^{-8}	7.17×10^{-35}	1.30×10^{-34}	0.6149	8.48×10^{-39}	1.57×10^{-12}	2.70×10^{-228}
	std	2.23×10^{-15}	7.45×10^{-17}	5.18×10^{-24}	6.69×10^{-50}	1.91×10^{-50}	3.25×10^{-15}	5.92×10^{-41}	1.42×10^{-12}	0
F3	Ave	1536.896	589.492	279.3439	2.75×10^{-15}	7.41×10^{-15}	6.91×10^{-7}	1.15×10^{-21}	0.0864	1.86×10^{-52}
	std	6.61×10^{-13}	7.12×10^{-13}	1.21×10^{-13}	2.65×10^{-31}	5.64×10^{-30}	6.79×10^{-25}	6.70×10^{-21}	0.1444	3.35×10^{-50}
F4	Ave	2.0942	3.9634	3.25×10^{-9}	9.42×10^{-15}	1.26×10^{-14}	6.13×10^{-4}	1.33×10^{-23}	2.60×10^{-8}	8.8×10^{-152}
	std	2.23×10^{-15}	1.99×10^{-16}	2.03×10^{-24}	2.12×10^{-30}	1.06×10^{-29}	6.20×10^{-22}	1.15×10^{-22}	9.25×10^{-9}	0
F5	Ave	310.4273	50.26245	36.10695	146.4564	26.8607	45.3518	28.8615	46.049	24.25997
	std	2.10×10^{-13}	1.59×10^{-14}	3.10×10^{-14}	1.91×10^{-14}	0	5.16×10^{-12}	4.76×10^{-3}	0.4219	1.51×10^{-14}
F6	Ave	14.55	20.25	0	0.4435	0.6423	6.14×10^{-8}	7.10×10^{-21}	0.398	0
	std	3.18×10^{-15}	7.56×10^{-4}	0	4.22×10^{-16}	6.21×10^{-17}	1.32×10^{-24}	1.12×10^{-25}	0.1914	0
F7	Ave	5.68×10^{-3}	0.1134	0.0206	0.0017	0.0008	0.7261	3.72×10^{-4}	0.0018	1.47×10^{-4}
	std	7.76×10^{-19}	4.34×10^{-17}	2.72×10^{-18}	3.88×10^{-19}	7.27×10^{-20}	4.93×10^{-16}	5.09×10^{-5}	0.001	5.45×10^{-20}

Analysis and comparison of the results obtained from optimization algorithms, shows that the SLOA has a more effective ability to provide quasi-optimal solutions in this type of objective functions than similar algorithms.

4.2. Evaluation of High-Dimensional Multimodal Objective Functions

The second group of objective functions, including six F8 to F13 functions of the high-dimension multi-objective type, are selected to analyze the power of optimization algorithms in solving this type of optimization problems. The performance results of the optimization algorithms and the SLOA on the objective functions F8 to F13 are presented in Table 3. The results of this table show that the proposed algorithm has been able to provide the global optimal solution for F9 and F11 functions. The proposed SLOA is the first best optimizer for solving F10 and F12 functions. In optimizing the F8 function, SLOA is the sixth best optimizer after GA, TLBO, PSO, GWO, and TSA algorithms. GSA is the first best optimizer for optimizing the F13 function, while the proposed algorithm for solving this objective function is the second-best optimizer.

Table 3. Optimization results of SLOA and other algorithms on high dimensional test function.

		GA	PSO	GSA	TLBO	GWO	GOA	TSA	MPA	SLOA
F8	Ave	-8184.41	-6908.6558	-2849.0724	-7408.6107	-5885.1172	-1783.2648	-5740.3388	-3594.16321	-5583.06
	std	$8.33 \times 10^{+2}$	$6.26 \times 10^{+2}$	$2.64 \times 10^{+2}$	$5.14 \times 10^{+2}$	$4.68 \times 10^{+2}$	$7.62 \times 10^{+2}$	$4.15 \times 10^{+1}$	$8.11 \times 10^{+2}$	2.03×10^{-12}
F9	Ave	62.4114	57.0613	$1.63 \times 10^{+1}$	$1.02 \times 10^{+1}$	8.53×10^{-15}	3.1962	5.70×10^{-3}	$1.40 \times 10^{+2}$	0
	std	2.54×10^{-14}	6.36×10^{-15}	3.18×10^{-15}	5.56×10^{-15}	5.64×10^{-30}	1.70×10^{-14}	1.46×10^{-3}	$2.63 \times 10^{+1}$	0
F10	Ave	3.2218	2.1546	3.57×10^{-9}	2.76×10^{-1}	1.71×10^{-14}	3.16×10^{-1}	9.80×10^{-14}	9.70×10^{-12}	4.44×10^{-15}
	std	5.16×10^{-15}	7.94×10^{-16}	3.70×10^{-25}	2.56×10^{-15}	2.75×10^{-29}	6.35×10^{-14}	4.51×10^{-12}	6.13×10^{-12}	0
F11	Ave	1.2302	0.0462	3.74	6.08×10^{-1}	3.70×10^{-3}	1.26×10^{-1}	1.00×10^{-7}	0	0
	std	8.44×10^{-16}	3.10×10^{-18}	2.78×10^{-15}	1.99×10^{-16}	1.26×10^{-18}	5.62×10^{-16}	7.46×10^{-7}	0	0
F12	Ave	0.047	0.4806	0.0362	0.0203	0.0372	1.5263	0.0368	0.0851	0.010594
	std	4.65×10^{-17}	1.86×10^{-16}	6.21×10^{-17}	7.76×10^{-16}	4.34×10^{-17}	7.53×10^{-10}	1.55×10^{-2}	0.0052	2.21×10^{-17}
F13	Ave	1.2085	0.5084	0.002	0.3293	0.5763	3.52×10^{-1}	2.96	0.4901	0.111886
	std	3.23×10^{-14}	4.97×10^{-15}	4.26×10^{-14}	2.11×10^{-14}	2.48×10^{-15}	6.46×10^{-12}	1.57×10^{-12}	0.1932	1.55×10^{-16}

Analysis of the results of optimization of high-dimensional multi-model objective functions shows the acceptable ability of the SLOA to solve such optimization problems.

4.3. Evaluation of Fixed-Dimensional Multimodal Objective Functions

Ten objective functions, including F14 to F23, are considered to analyze the performance of optimization algorithms in solving fixed-dimension multimodal optimization problems. The results of optimization of these objective functions using the proposed SLOA and eight other algorithms are presented in Table 4. The optimization results indicate that the proposed SLOA is the first best optimizer for the F15, F16, F17, F19, and F20 functions. In optimizing the F14 function, SLOA has a similar performance to GA, GOA, and MPA in the Ave indicator. However, due to the lower std indicator, it is clear that the proposed SLOA is a more efficient method for solving F14. In optimizing the F18 function, SLOA with the lower std indicator is the first best optimizer. In optimizing the F21, F22, and F23 functions, SLOA and MPA provide similar performance in the Ave indicator. But since the proposed SLOA has a lower std indicator, it is the first-best optimizer and MPA is the second-best optimizer.

Table 4. Optimization results of SLOA and other algorithms on fixed dimensional test function.

		GA	PSO	GSA	TLBO	GWO	GOA	TSA	MPA	SLOA
F14	Ave	9.99×10^{-1}	2.17	3.59	2.27	3.74	9.98×10^{-1}	1.99	0.998	0.998
	std	1.56×10^{-15}	7.94×10^{-16}	7.94×10^{-16}	1.99×10^{-16}	6.45×10^{-15}	7.62×10^{-15}	2.65×10^{-7}	4.27×10^{-16}	1.47×10^{-16}
F15	Ave	5.40×10^{-2}	5.35×10^{-2}	2.40×10^{-3}	3.30×10^{-3}	0.0063	5.10×10^{-3}	4.00×10^{-4}	3.00×10^{-3}	0.0003
	std	7.08×10^{-18}	3.88×10^{-16}	2.91×10^{-18}	1.22×10^{-17}	1.16×10^{-18}	2.17×10^{-16}	9.0125×10^{-4}	4.10×10^{-15}	7.15×10^{-19}
F16	Ave	-1.0316	-1.03	-1.03	-1.03	-1.03	-1.03	-1.03	-1.03	-1.03163
	std	7.94×10^{-16}	3.48×10^{-16}	5.96×10^{-16}	1.44×10^{-15}	3.97×10^{-16}	6.33×10^{-15}	5.65×10^{-16}	4.47×10^{-16}	2.48×10^{-16}
F17	Ave	0.4369	7.85×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	4.04×10^{-1}	3.99×10^{-1}	3.98×10^{-1}	0.397887
	std	4.97×10^{-14}	4.97×10^{-15}	9.93×10^{-16}	7.45×10^{-16}	8.69×10^{-16}	3.83×10^{-14}	2.16×10^{-16}	9.12×10^{-15}	0
F18	Ave	4.3592	3	3	3.0009	3	3	3	3	3
	std	5.96×10^{-16}	3.67×10^{-15}	6.95×10^{-16}	1.59×10^{-15}	2.09×10^{-15}	8.61×10^{-15}	2.65×10^{-15}	1.96×10^{-15}	1.94×10^{-17}
F19	Ave	-3.85434	-3.8627	-3.8627	-3.8609	-3.86	-3.81	-3.8066	-3.8627	-3.86278
	std	9.93×10^{-14}	8.94×10^{-15}	8.34×10^{-15}	7.35×10^{-15}	2.48×10^{-15}	7.62×10^{-15}	2.64×10^{-15}	4.24×10^{-15}	5.96×10^{-16}
F20	Ave	-2.8239	-3.2619	-3.0396	-3.2014	-3.2523	-3.24	-3.3206	-3.32	-3.32199
	std	3.97×10^{-11}	2.98×10^{-12}	2.18×10^{-14}	1.79×10^{-15}	2.18×10^{-15}	5.68×10^{-15}	5.69×10^{-15}	1.14×10^{-11}	1.99×10^{-16}
F21	Ave	-4.304	-5.3891	-5.1486	-9.1746	-9.6452	-7.3862	-5.5021	-10.1532	-10.1532
	std	1.59×10^{-12}	1.49×10^{-13}	2.98×10^{-14}	8.54×10^{-15}	6.55×10^{-15}	5.61×10^{-10}	5.46×10^{-13}	2.54×10^{-11}	7.94×10^{-16}
F22	Ave	-5.12	-7.63	-9.02	$-1.00 \times 10^{+1}$	$-1.04 \times 10^{+1}$	-8.81	-5.0625	-10.4029	-10.4029
	std	6.29×10^{-15}	7.59×10^{-15}	1.65×10^{-12}	1.53×10^{-14}	1.99×10^{-15}	8.42×10^{-14}	8.46×10^{-14}	2.82×10^{-11}	6.36×10^{-16}
F23	Ave	-6.56	-6.16	-8.90	-9.29	$-1.01 \times 10^{+1}$	-9.96	-10.3613	-10.5364	-10.5364
	std	3.87×10^{-15}	2.78×10^{-15}	7.15×10^{-14}	6.19×10^{-15}	4.57×10^{-15}	8.61×10^{-14}	7.65×10^{-12}	3.99×10^{-11}	1.45×10^{-16}

What is clear from the analysis and comparison of the performance of optimization algorithms is that the SLOA has a high ability to solve fixed-dimensional multimodal optimization problems and is able to provide more effective solutions with less standard deviation than similar algorithms.

The behavior and performance of the proposed SLOA and the eight compared optimization algorithms are presented in the form of a boxplot in Figure 2. This figure intuitively demonstrates the superiority of the proposed SLOA in optimizing the functions of F1 to F7, F9 to F12, F14 to F23.

4.4. Statistic Analysis

Presenting the results of optimization of objective functions using standard statistical indicators of average and standard deviation of the best solutions provides useful and valuable information. However, the superiority of an algorithm in solving an optimization problem may be coincidental even after twenty independent implementations. Therefore, Wilcoxon rank-sum test is presented in order to statistically analyze the optimization results obtained from the proposed SLOA and eight other optimization algorithms. Wilcoxon rank-sum test is a non-parametric test that is used to evaluate the similarity of two dependent samples with a ranking scale. This analysis is applied to specify whether the obtained results using SLOA are different from the competitive algorithms in a statistically significant way.

A *p*-value characterizes whether the given optimization algorithm is statistically significant or not. If the *p*-value of the given optimization algorithm is less than 0.05, then the corresponding optimization algorithm is statistically significant. The results of statistical analysis based on the application of Wilcoxon rank-sum test are presented in

Table 5. Based on the analysis of the results in Table 5, it is concluded that the proposed SLOA has a significant superiority over the other eight algorithms in cases where the p -value is less than 0.05. Accordingly, the proposed SLOA has a significant superiority over all eight compared algorithms in optimizing the F1 to F7 unimodal functions. In optimizing high-dimensional multimodal functions, SLOA has a significant superiority over MPA and GOA with p -value of less than 0.05. Also, in optimizing fixed-dimensional multimodal functions, the proposed SLOA with p -value of less than 0.05 has a significant superiority over all eight algorithms compared.

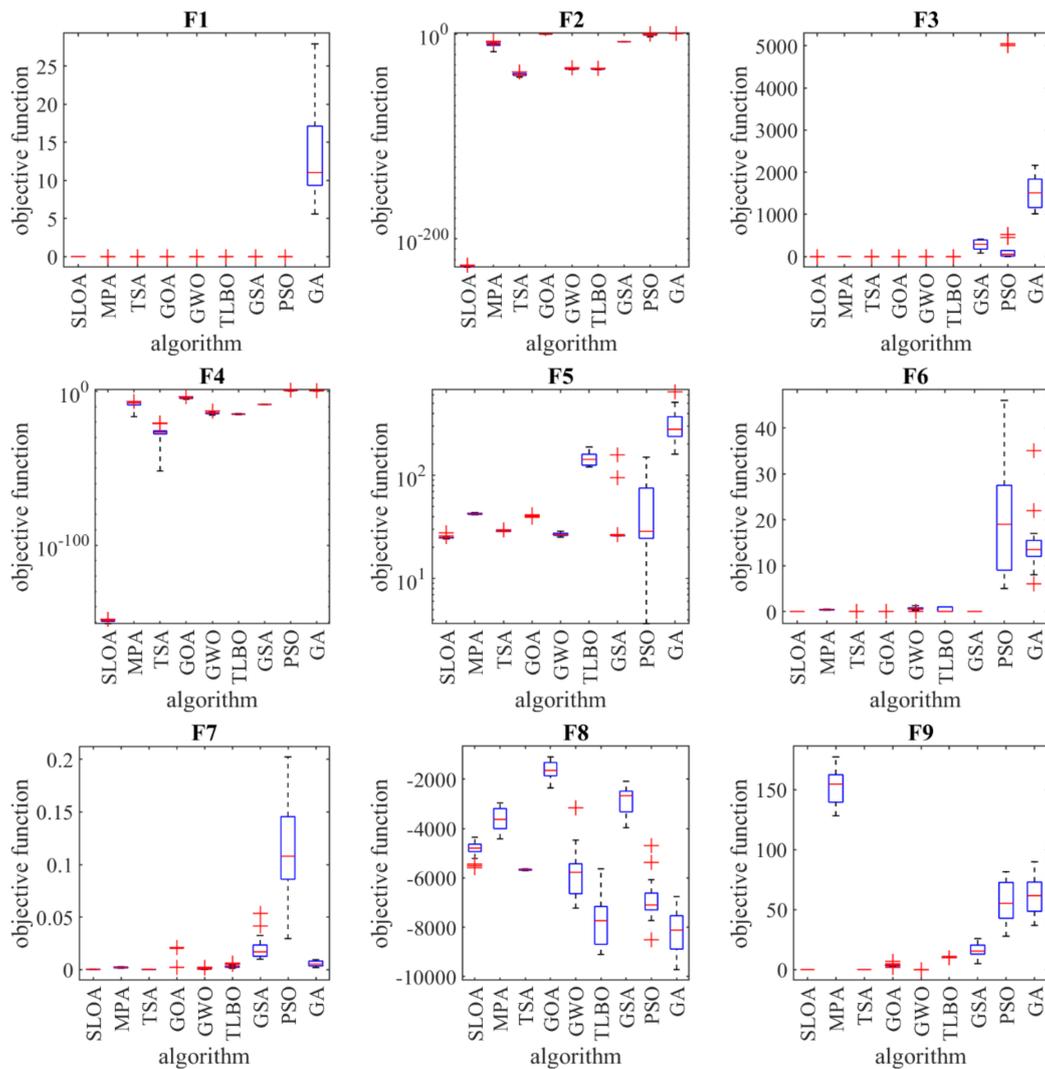


Figure 2. Cont.

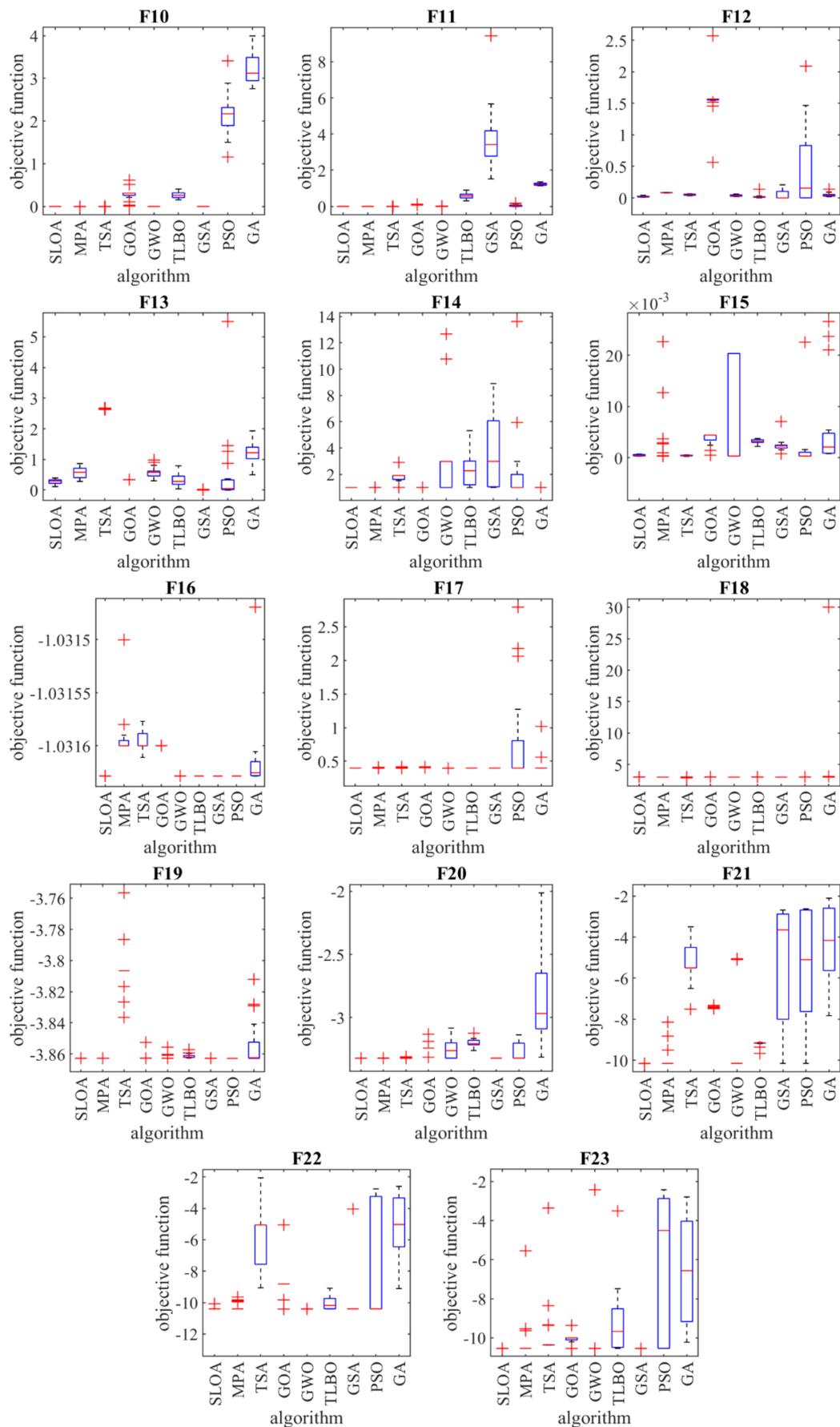


Figure 2. Boxplot of composition objective functions results for different optimization algorithms.

Table 5. Obtained results from the Wilcoxon test ($p \geq 0.05$).

Compared Algorithms	Unimodal	High—Multimodal	Fixed—Multimodal
SLOA vs. MPA	0.015625	0.0625	0.0625
SLOA vs. TSA	0.015625	0.4375	0.003906
SLOA vs. GOA	0.015625	0.03125	0.007813
SLOA vs. GWO	0.015625	0.4375	0.011719
SLOA vs. TLBO	0.015625	0.4375	0.005859
SLOA vs. GSA	0.03125	0.15625	0.019531
SLOA vs. PSO	0.015625	0.4375	0.003906
SLOA vs. GA	0.015625	0.4375	0.001953

4.5. Sensitivity Analysis

The sensitivity analysis means studies of the output changes of a mathematical model due to changes in the values of the input parameters. In other words, if sensitivity analysis is checked, which if the value of an independent parameter is changed, how does its dependent variable change in a specified and defined condition with assuming the constant of other parameters? In order to further analyze the proposed SLOA, sensitivity analysis is presented. For this purpose, the sensitivity of the SLOA to the three parameters of maximum number of iterations, number of members of the snow leopard population, and P parameter is investigated.

In order to analyze the sensitivity of the proposed algorithm to the maximum number of iterations parameter, the SLOA in independent runs for a maximum of 100, 500, 800, and 1000 iterations is implemented on the objective functions F1 to F23. The results of this analysis are presented in Table 6 and the behavior of convergence curves under the influence of sensitivity analysis to the maximum number of iterations are presented in Figure 3. The simulation results show that by increasing the maximum number of iterations, SLOA converges towards more suitable quasi-optimal solutions.

The proposed SLOA is implemented in independent runs for the population of snow leopards 20, 30, 50, and 80 on all objective functions F1 to F23 in order to analyze the sensitivity of the SLOA to the parameter of the number of members of the population. The simulation results of the sensitivity analysis to the number of population members are presented in Table 7, and the behavior of the convergence curves is shown in Figure 4. The simulation results show that as the number of snow leopard population members increases, the values of the objective function decrease and the algorithm converges towards solutions closer to the global optimal.

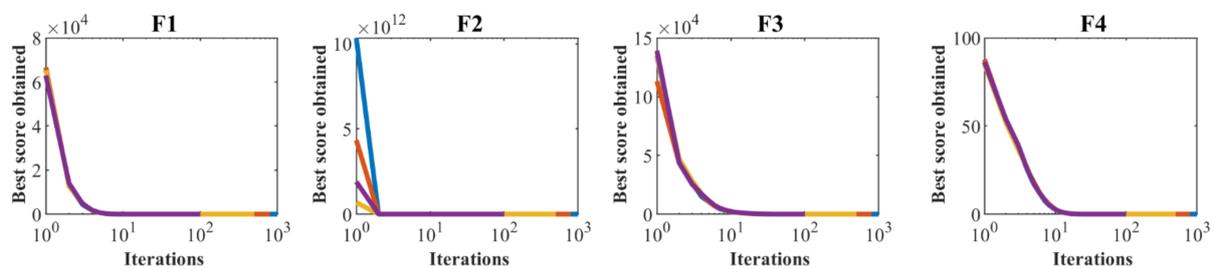


Figure 3. Cont.

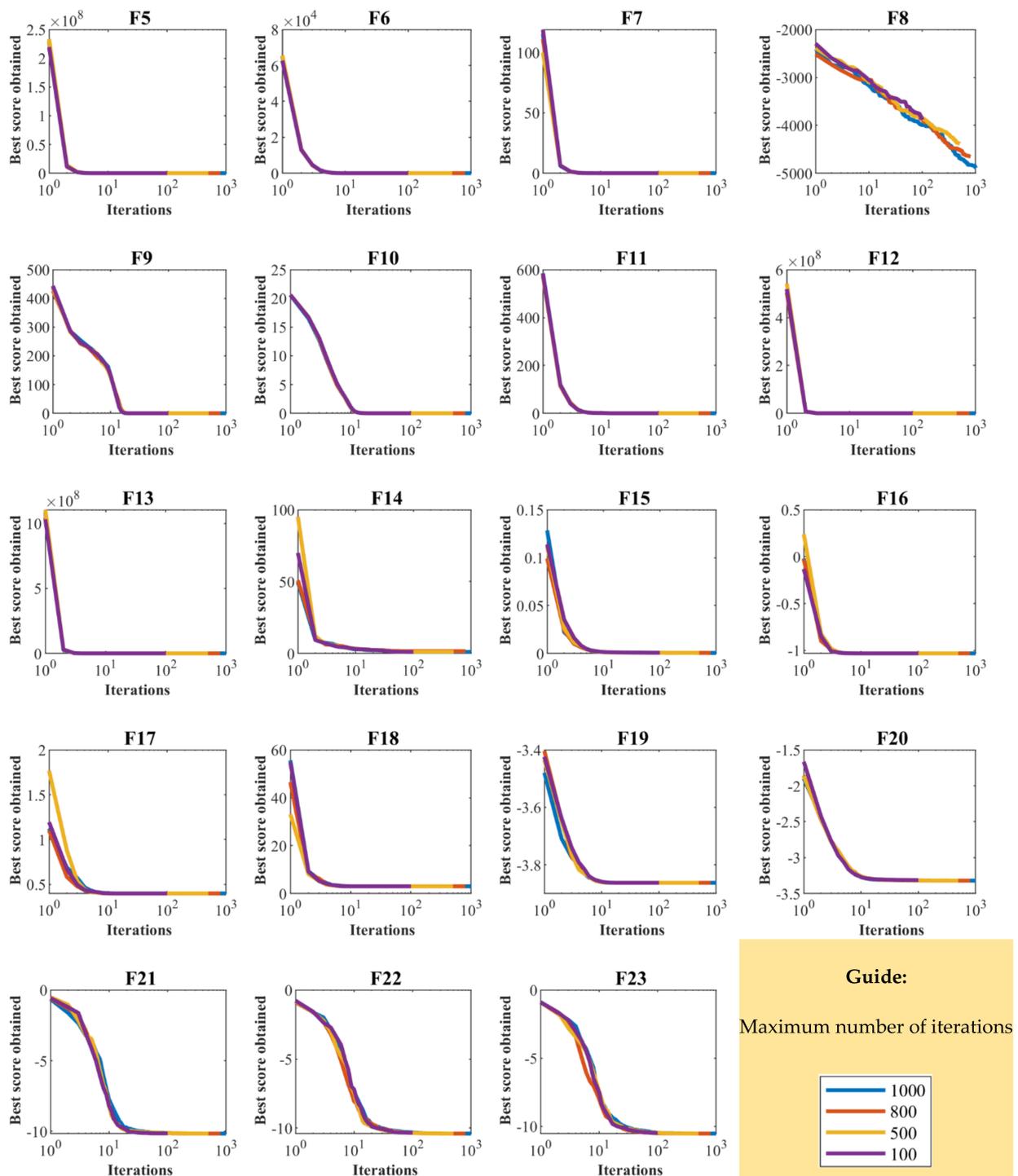


Figure 3. Sensitivity analysis of SLOA for maximum number of iterations.

Table 6. Results of the algorithm sensitivity analysis to the maximum number of iterations.

Objective Function	Maximum Number of Iterations			
	100	500	800	1000
F ₁	4.74×10^{-39}	3×10^{-213}	0	0
F ₂	2.42×10^{-21}	1×10^{-112}	3.1×10^{-181}	2.7×10^{-228}
F ₃	0.030659	8.02×10^{-18}	1.14×10^{-28}	1.86×10^{-52}
F ₄	6.24×10^{-14}	3.96×10^{-74}	2.8×10^{-119}	8.8×10^{-152}
F ₅	26.82582	25.78279	25.44565	24.25997
F ₆	0	0	0	0
F ₇	0.002388	0.000623	0.000463	0.000147
F ₈	-3883.26	-4381.65	-4649.31	-5583.06
F ₉	0	0	0	0
F ₁₀	4.62×10^{-15}	4.44×10^{-15}	4.44×10^{-15}	4.44×10^{-15}
F ₁₁	0	0	0	0
F ₁₂	0.035171	0.030258	0.024427	0.010594
F ₁₃	0.376541	0.349986	0.269467	0.111886
F ₁₄	0.998008	1.097209	0.998	0.998
F ₁₅	0.00059	0.000538	0.000446	0.0003
F ₁₆	-1.03163	-1.03163	-1.03163	-1.03163
F ₁₇	0.397887	0.397887	0.397887	0.397887
F ₁₈	3	3	3	3
F ₁₉	-3.86278	-3.86278	-3.86278	-3.86278
F ₂₀	-3.31457	-3.32195	-3.32196	-3.32199
F ₂₁	-10.1393	-10.1521	-10.1528	-10.1532
F ₂₂	-10.35	-10.3681	-10.4022	-10.4029
F ₂₃	-10.5031	-10.5357	-10.5361	-10.5364

The P parameter represents the percentage of the distance that the snow leopard walks when attacking prey. In order to analyze the sensitivity of the proposed algorithm to this parameter, SLOA is run independently for different p values equal to 0.2, 0.375, 0.6, and 0.8. The simulation results of this analysis for all F1 to F23 objective functions are reported in Table 8 and the behavior of convergence curves under the influence of sensitivity analysis to the P parameter are presented in Figure 5. The simulation results show that in the target objective functions of F6, F9, F11, F14, F16, F17, F18, F19, F20, and F23, the P parameter changes had no effect on the performance of the proposed SLOA. In the F1, F5, F10, F21, and F22 functions, the P parameter changes had very little effect on the performance of the proposed SLOA. According to the analysis of the optimization results, it is determined that the value of 0.375 for the P parameter is a suitable value. Therefore, the authors suggest that researchers use the value 0.375 for the P parameter in their research simulations.

Table 7. Results of the algorithm sensitivity analysis to the number of population members.

Objective Function	Number of Population Members			
	20	30	50	80
F ₁	0	0	0	0
F ₂	2.8×10^{-223}	7.2×10^{-225}	2.7×10^{-228}	9.6×10^{-230}
F ₃	1.38×10^{-41}	1.92×10^{-51}	1.86×10^{-52}	7.69×10^{-62}
F ₄	2.2×10^{-146}	2×10^{-149}	8.8×10^{-152}	2.6×10^{-155}
F ₅	26.65993	26.17808	24.25997	24.12891
F ₆	0	0	0	0
F ₇	0.001409	0.000883	0.000147	9.81×10^{-5}
F ₈	-4525.52	-4571.32	-5583.06	-5894.9
F ₉	0	0	0	0
F ₁₀	5.15×10^{-15}	4.44×10^{-15}	4.44×10^{-15}	4.44×10^{-15}
F ₁₁	0.001355	0	0	0
F ₁₂	0.115373	0.054686	0.010594	0.00869
F ₁₃	1.318733	0.699014	0.111886	0.053976
F ₁₄	2.760396	2.272137	0.998	0.998
F ₁₅	0.00156	0.000499	0.0003	0.0003
F ₁₆	-1.03163	-1.03163	-1.03163	-1.03163
F ₁₇	0.397916	0.39789	0.397887	0.397887
F ₁₈	7.05	3	3	3
F ₁₉	-3.86275	-3.86278	-3.86278	-3.86278
F ₂₀	-3.29751	-3.29803	-3.32199	-3.32199
F ₂₁	-9.54574	-10.1514	-10.1532	-10.1532
F ₂₂	-8.61249	-10.1353	-10.4029	-10.4029
F ₂₃	-8.21416	-10.213	-10.5364	-10.5364

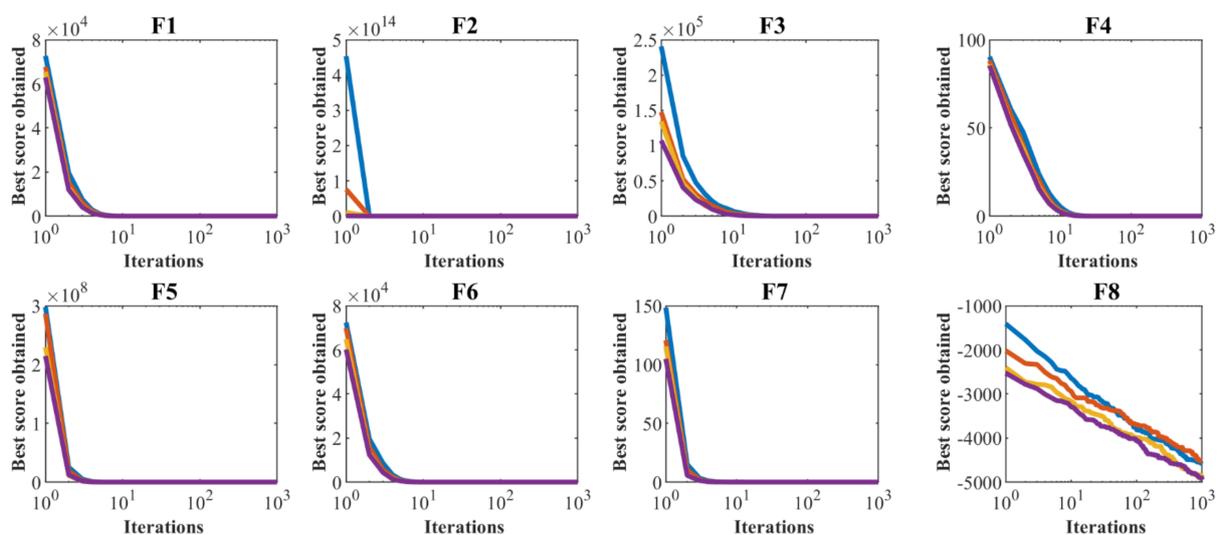


Figure 4. Cont.

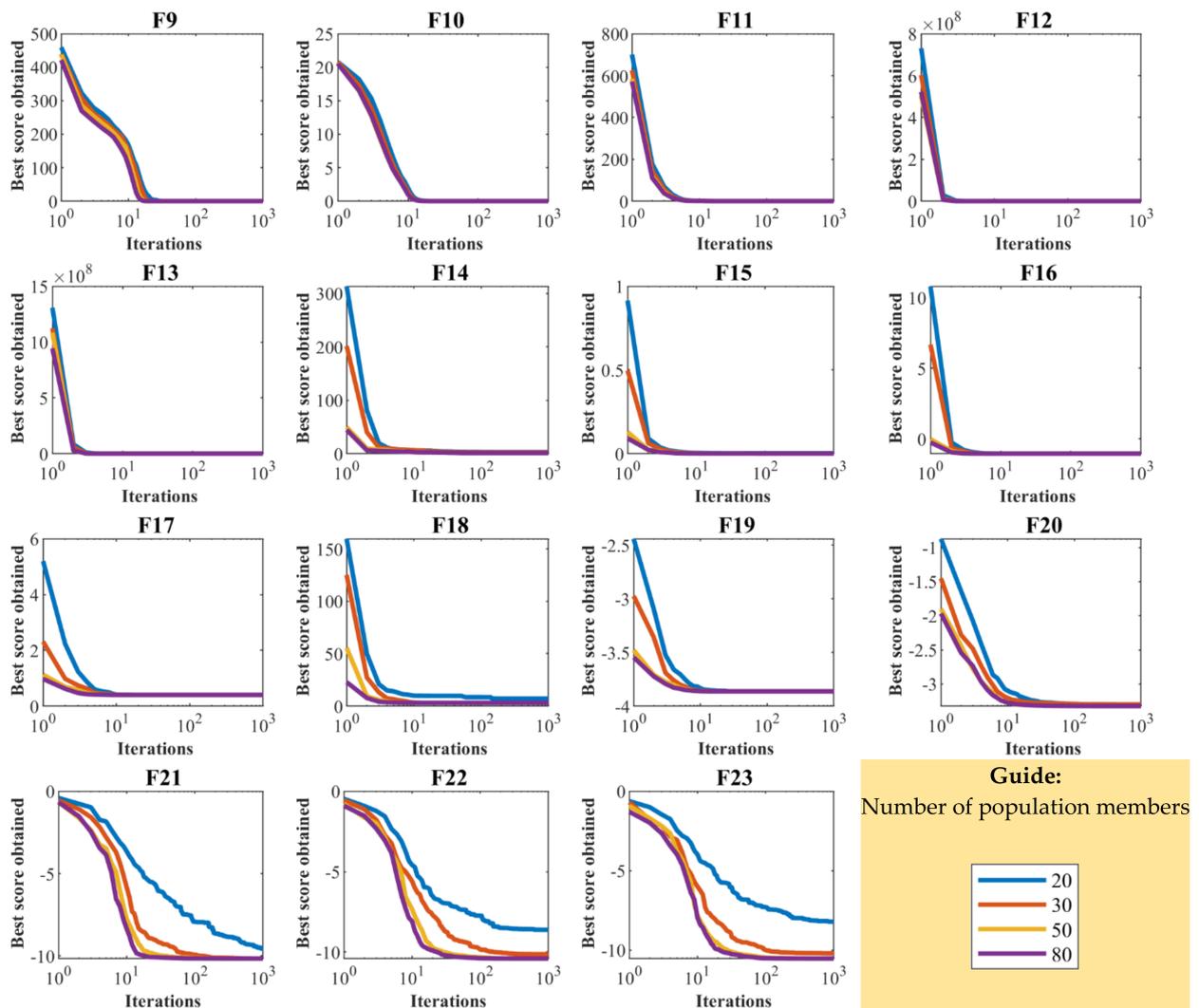


Figure 4. Sensitivity analysis of SLOA for number of population members.

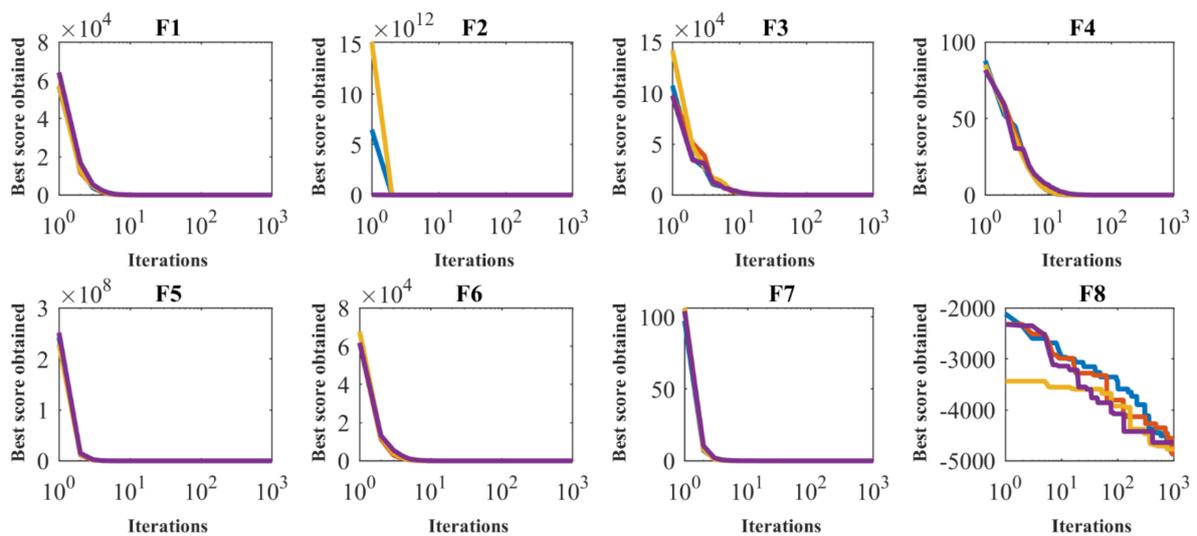


Figure 5. Cont.

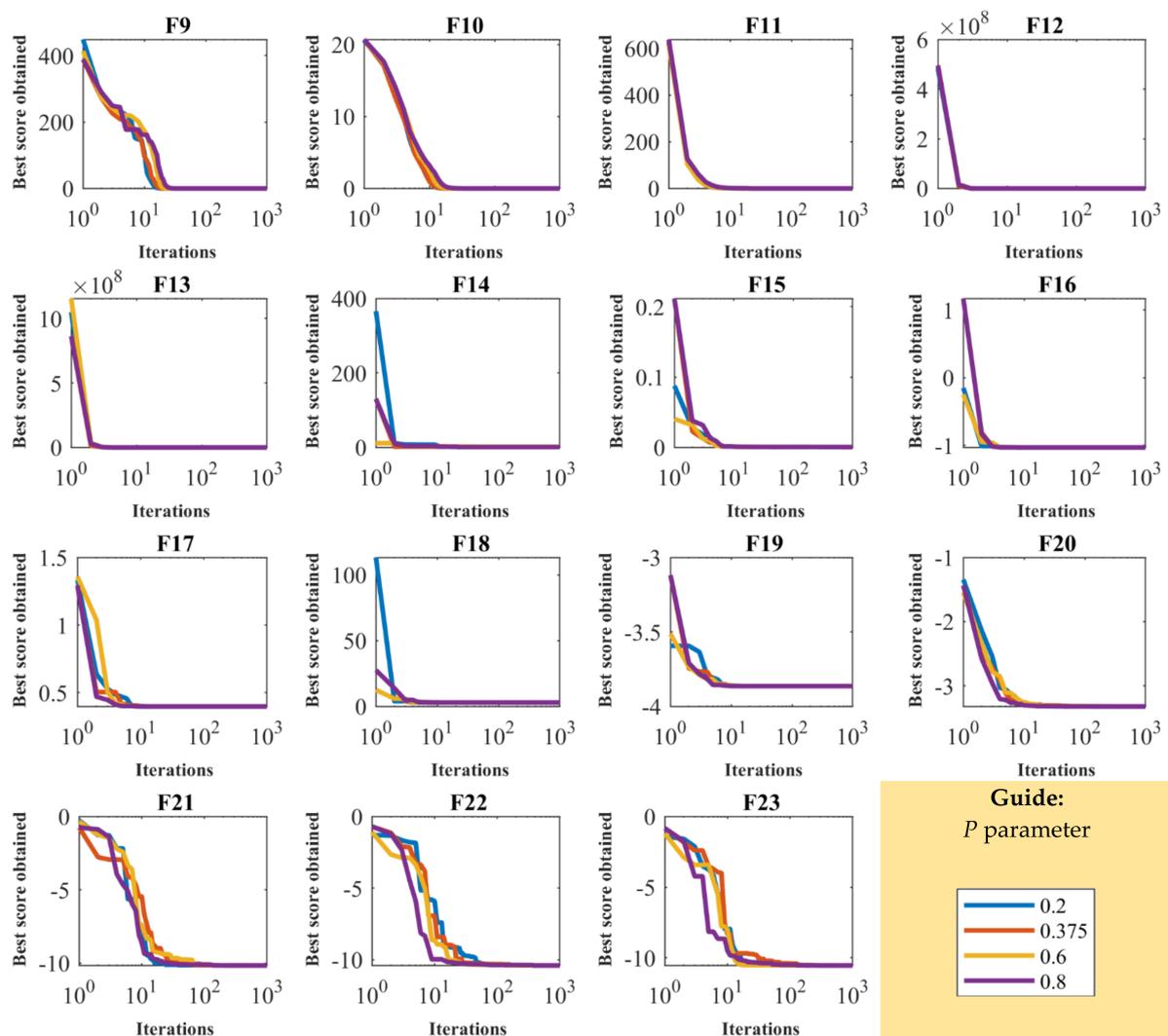


Figure 5. Sensitivity analysis of SLOA for P parameter.

Table 8. Results of the algorithm sensitivity analysis to the number of population members.

Objective Function	Value of P Parameter			
	0.2	0.375	0.6	0.8
F ₁	0	0	0	3.7×10^{-218}
F ₂	1.5×10^{-230}	2.7×10^{-228}	1.8×10^{-187}	1.2×10^{-128}
F ₃	6.71×10^{-44}	1.86×10^{-52}	1.76×10^{-36}	4.61×10^{-28}
F ₄	1.7×10^{-164}	8.8×10^{-152}	1.43×10^{-94}	1.38×10^{-50}
F ₅	24.82493	24.25997	24.48763	24.33416
F ₆	0	0	0	0
F ₇	0.000232	0.000147	0.000272	0.000438
F ₈	-4851.53	-5583.06	-4864.46	-4643.05
F ₉	0	0	0	0
F ₁₀	4.44×10^{-15}	4.44×10^{-15}	4.44×10^{-15}	7.99×10^{-15}
F ₁₁	0	0	0	0
F ₁₂	0.006463	0.010594	0.021812	0.019961

Table 8. Cont.

Objective Function	Value of P Parameter			
	0.2	0.375	0.6	0.8
F ₁₃	0.23632	0.111886	0.225848	0.627989
F ₁₄	0.998	0.998	0.998	0.998
F ₁₅	0.000399	0.0003	0.000341	0.00037
F ₁₆	−1.03163	−1.03163	−1.03163	−1.03163
F ₁₇	0.397887	0.397887	0.397887	0.397887
F ₁₈	3	3	3	3
F ₁₉	−3.86278	−3.86278	−3.86278	−3.86278
F ₂₀	−3.32199	−3.32199	−3.32199	−3.32199
F ₂₁	−10.1525	−10.1532	−10.1528	−10.1532
F ₂₂	−10.4029	−10.4029	−10.4028	−10.4027
F ₂₃	−10.5364	−10.5364	−10.5364	−10.5364

5. Discussion

Exploitation and exploration are two important and key criteria in analyzing the performance of optimization algorithms and evaluating their ability to provide appropriate quasi-optimal solutions.

Exploitation means the ability of the algorithm to provide a suitable quasi-optimal solution that is close to the global optimal. Therefore, compared to the ability of several algorithms to solve an optimization problem, the algorithm that can provide the best quasi-optimal solution and the closest to the global optimal has a higher exploitation. Exploitation is very important especially for optimization problems that lack local optimal solutions. Unimodal objective functions, including F1 to F7, have only one main solution and lack local optimal solutions. Therefore, these types of objective functions are suitable for evaluating the exploitation of optimization algorithms. The results of optimization of these objective functions are presented in Table 2. Based on the analysis of the results of this table and the comparison of the performance of optimization algorithms, it is clear that the proposed SLOA has provided more suitable quasi-optimal solutions and has a higher capability in the exploitation index than the other eight algorithms.

Exploration means the ability of the algorithm to accurately and effectively scan the search space of optimization problems. This index is especially important in solving optimization problems that have several local optimal solutions in addition to the global optimal. An optimization algorithm must be able to search the various regions of the problem-solving space well during execution and approach the global optimal by passing through the local optimal regions. Therefore, in evaluating the performance of several algorithms in solving an optimization problem, an algorithm that can scan the problem-solving space well has a higher ability in the exploration index. High-dimensional multimodal objective functions, including F8 to F13, as well as fixed-dimensional multimodal objective functions, including F14 to F23, are functions that have several optimal local solutions in addition to the main global optimal. Therefore, these types of objective functions are suitable for analyzing the exploration index in optimization algorithms. The performance results of optimization algorithms in solving these objective functions are presented in Tables 3 and 4. The analysis of these results indicates that the proposed SLOA with high exploration power has scanned the search space in optimization problems containing local optimal areas and has approached to global optimal by passing local areas.

Exploration power allows the algorithm to scan different areas of the search space with the aim of passing through the optimal local areas and discovering the main optimal area. Exploitation power causes an algorithm to converge as much as possible towards the optimal solution after finding the optimal area in the search space. The main feature

and advantage of the proposed SLOA is that in the process of scanning the search space, it does not rely on a specific member like the best member of the population. The first and second phases of SLOA, which simulate the snow leopard zigzag movement and hunting strategy, increase the exploitation power of the proposed algorithm in converging to the optimal solution. The effect of these two phases and the high convergence power of the proposed SLOA in the results of optimizing the F1 to F4 functions is significantly evident. The third and fourth phases of SLOA, which models reproduction and mortality, have a great impact on the algorithm's exploration power in scanning new areas in the search space and distancing itself from non-optimal areas. The high exploration power of the proposed SLOA is well evident in the optimization of F9, F11, F14, to F23 functions.

Therefore, with an overview of the optimization results obtained from three different groups of objective functions, it has been determined that the proposed SLOA has high capability and power in both exploitation and exploration indicators. In fact, the main reason for the proposed algorithm's superiority compared to compared algorithms is its exploitation and exploration abilities. SLOA with high-exploration capability can search different areas of problem-solving space and after crossing local optimal areas and approaching the optimal global solution, with high-exploration capability, converge to global optimal as much as possible. As specified in simulations, SLOA in optimizing functions of F1, F6, F9, and F11 has been able to provide the global optimal solution. What is inferred from the simulation results is that the performance of the proposed SLOA in optimization is superior to the comparative algorithms, and its results are much more competitive.

6. Conclusions and Future Studies

Numerous optimization problems in different sciences should be optimized with appropriate technique. Population-based optimization algorithms are among the stochastic solving methods of optimization problems that can provide acceptable quasi-optimal solutions to optimization problems. In this paper, a new optimization algorithm called Snow Leopard Optimization Algorithm (SLOA) has been presented in order to effectively solve optimization problems in various sciences and provide quasi-optimal solutions that are more desirable and closer to the global optimal. The theory and different stages of the proposed SLOA have been stated and then the mathematical model of the SLOA has been presented in order to implement on optimization problems with the aim of achieving quasi-optimal solutions. The performance of the proposed SLOA has been tested on a standard set consisting of twenty-three objective functions of unimodal, high-dimensional multimodal, and fixed-dimensional multimodal types. Also, in order to analyze the quality of the SLOA in providing quasi-optimal solutions, the results obtained from the SLOA have been compared with eight other well-known algorithms, namely Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Gravitational Search Algorithm (GSA), Teaching-Learning Based Optimization (TLBO), Gray Wolf Optimizer (GWO), Grasshopper Optimization Algorithm (GOA), Marine Predators Algorithm (MPA), and Tunicate Swarm Algorithm (TSA).

The optimization results using the proposed algorithm indicate the high ability of the SLOA to provide suitable quasi-optimal solutions for optimization problems of different types. Also, the results of performance analysis and comparison of comparative optimization algorithms showed that the SLOA presented better results and is much more competitive than these eight optimization algorithms.

The authors offer several suggestions and perspectives for future studies of this paper. The design of binary and multi-objective versions for SLOA are the main potentials of the proposed algorithm. In addition, the use of the proposed SLOA in solving real-world optimization problems as well as various other types of optimization problems are suggestions for future studies related to this paper—see e.g., [34–39].

The important thing about all optimization algorithms is that one cannot claim that a particular algorithm is the best optimizer for all optimization problems. Therefore, one of the limitations of the proposed SLOA is that in optimizing some optimization problems, it may not be able to provide a quasi-optimal solution very close to the global optimal. Another limitation for SLOA is that it is always possible to design newer algorithms that have a higher ability to converge to the optimal solution. In addition, with the advancement of science and technology, more complex optimization issues arise that existing algorithms, such as the proposed algorithm, may not be able to solve and require the improvement of existing methods or the design of newer methods.

Author Contributions: Conceptualization, Š.H. and P.C.; methodology, P.C.; software, Š.H. and P.C.; validation, Š.H., P.C. and M.H.; formal analysis, Z.B.; investigation, M.H.; resources, Š.H.; data curation, P.C.; writing, Š.H., Z.B. and P.C. All authors have read and agreed to the published version of the manuscript.

Funding: The research was supported by the Project of Specific Research, PrF UHK No. 2101/2021 and by the project KEGA, 036UKF-4/2019, Adaptation of the learning process using sensor networks and the Internet of Things.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The authors declare to honor the Principles of Transparency and Best Practice in Scholarly Publishing about Data.

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

Appendix A

Information of the twenty-three objective functions is provided in Tables A1–A3.

Table A1. Unimodal test functions.

	Objective Function	Range	Dimensions	F_{\min}
1.	$F_1(x) = \sum_{i=1}^m x_i^2$	$[-100, 100]$	30	0
2.	$F_2(x) = \sum_{i=1}^m x_i + \prod_{i=1}^m x_i $	$[-10, 10]$	30	0
3.	$F_3(x) = \sum_{i=1}^m \left(\sum_{j=1}^i x_j \right)^2$	$[-100, 100]$	30	0
4.	$F_4(x) = \max\{ x_i , 1 \leq i \leq m\}$	$[-100, 100]$	30	0
5.	$F_5(x) = \sum_{i=1}^{m-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$	$[-30, 30]$	30	0
6.	$F_6(x) = \sum_{i=1}^m ([x_i + 0.5])^2$	$[-100, 100]$	30	0
7.	$F_7(x) = \sum_{i=1}^m ix_i^4 + \text{random}(0, 1)$	$[-1.28, 1.28]$	30	0

Table A2. High dimensional Multimodal test functions.

	Objective Function	Range	Dimensions	F _{min}
8.	$F_8(x) = \sum_{i=1}^m -x_i \sin(\sqrt{ x_i })$	[-500, 500]	30	$-1.2569 \times 10^{+4}$
9.	$F_9(x) = \sum_{i=1}^m [x_i^2 - 10 \cos(2\pi x_i) + 10]$	[-5.12, 5.12]	30	0
10.	$F_{10}(x) = -20 \exp\left(-0.2\sqrt{\frac{1}{m} \sum_{i=1}^m x_i^2}\right) - \exp\left(\frac{1}{m} \sum_{i=1}^m \cos(2\pi x_i)\right) + 20 + e$	[-32, 32]	30	0
11.	$F_{11}(x) = \frac{1}{4000} \sum_{i=1}^m x_i^2 - \prod_{i=1}^m \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-600, 600]	30	0
12.	$F_{12}(x) = \frac{\pi}{m} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^m (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\}$ $+ \sum_{i=1}^m u(x_i, 10, 100, 4)$ $u(x_i, a, i, n) = \begin{cases} k(x_i - a)^n, & x_i > -a; \\ 0, & -a < x_i < a; \\ k(-x_i - a)^n, & x_i < -a. \end{cases}$	[-50, 50]	30	0
13.	$F_{13}(x) = 0.1 \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^m (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_m)] \right\} + \sum_{i=1}^m u(x_i, 5, 100, 4)$	[-50, 50]	30	0

Table A3. Fixed dimensional Multimodal test functions.

	Objective Function	Range	Dimensions	F _{min}
14.	$F_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right)^{-1}$	[-65.53, 65.53]	2	0.998
15.	$F_{15}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_i(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	[-5, 5]	4	0.00030
16.	$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	[-5, 5]	2	-1.0316
17.	$F_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos x_1 + 10$	[-5, 10] × [0, 15]	2	0.398
18.	$F_{18}(x) = [1, +, (x_1 + x_2 + 1)^2, (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2) + 3x_2^2] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1) + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2]$	[-5, 5]	2	3
19.	$F_{19}(x) = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^3 a_{ij} (x_j - P_{ij})^2\right)$	[0, 1]	3	-3.86
20.	$F_{20}(x) = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^6 a_{ij} (x_j - P_{ij})^2\right)$	[0, 1]	6	-3.22
21.	$F_{21}(x) = -\sum_{i=1}^5 [(X - a_i)(X - a_i)^T + 6c_i]^{-1}$	[0, 10]	4	-10.1532
22.	$F_{22}(x) = -\sum_{i=1}^7 [(X - a_i)(X - a_i)^T + 6c_i]^{-1}$	[0, 10]	4	-10.4029
23.	$F_{23}(x) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + 6c_i]^{-1}$	[0, 10]	4	-10.5364

References

1. Dhiman, G. SSC: A hybrid nature-inspired meta-heuristic optimization algorithm for engineering applications. *Knowl.-Based Syst.* **2021**, *222*, 106926. [CrossRef]
2. Dhiman, G.; Kumar, V. Emperor penguin optimizer: A bio-inspired algorithm for engineering problems. *Knowl.-Based Syst.* **2018**, *159*, 20–50. [CrossRef]
3. Doumari, S.A.; Givi, H.; Dehghani, M.; Malik, O.P. Ring Toss Game-Based Optimization Algorithm for Solving Various Optimization Problems. *Int. J. Intell. Eng. Syst.* **2021**, *14*, 545–554.
4. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [CrossRef]

5. Dehghani, M.; Montazeri, Z.; Dehghani, A.; Ramirez-Mendoza, R.A.; Samet, H.; Guerrero, J.M.; Dhiman, G. MLO: Multi leader optimizer. *Int. J. Intell. Eng. Syst.* **2020**, *13*, 364–373. [[CrossRef](#)]
6. Yang, X.-S. Firefly algorithm, stochastic test functions and design optimisation. *Int. J. Bio-Inspired Comput.* **2010**, *2*, 78–84. [[CrossRef](#)]
7. Yang, X.S.; Gandomi, A.H. Bat algorithm: A novel approach for global engineering optimization. *Eng. Comput.* **2012**, *29*, 464–483. [[CrossRef](#)]
8. Dhiman, G.; Kaur, A. STOA: A bio-inspired based optimization algorithm for industrial engineering problems. *Eng. Appl. Artif. Intell.* **2019**, *82*, 148–174. [[CrossRef](#)]
9. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl.-Based Syst.* **2015**, *89*, 228–249. [[CrossRef](#)]
10. Geem, Z.W.; Kim, J.H.; Loganathan, G.V. A new heuristic optimization algorithm: Harmony search. *Simulation* **2001**, *76*, 60–68. [[CrossRef](#)]
11. Dhiman, G.; Kumar, V. Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications. *Adv. Eng. Softw.* **2017**, *114*, 48–70. [[CrossRef](#)]
12. Sadeghi, A.; Doumari, S.A.; Dehghani, M.; Montazeri, Z.; Trojovský, P.; Ashtiani, H.J. A New “Good and Bad Groups-Based Optimizer” for Solving Various Optimization Problems. *Appl. Sci.* **2021**, *11*, 4382. [[CrossRef](#)]
13. Yilmaz, A.E.; Kuzuoglu, M. Calculation of optimized parameters of rectangular microstrip patch antenna using particle swarm optimization. *Microw. Opt. Technol. Lett.* **2007**, *49*, 2905–2907. [[CrossRef](#)]
14. Duca, A.; Duca, L.; Ciuprina, G.; Egemen Yilmaz, A.; Altinoz, T. PSO algorithms and GPGPU technique for electromagnetic problems. *Int. J. Appl. Electromagn. Mech.* **2017**, *53* (Suppl. S2), S249–S259. [[CrossRef](#)]
15. Altinoz, O.T.; Yilmaz, A.E.; Weber, G.-W. Application of chaos embedded PSO for PID parameter tuning. *Int. J. Comput. Commun.* **2012**, *7*, 204–217. [[CrossRef](#)]
16. Stanković, A.; Petrović, G.; Čojbašić, Ž.; Marković, D. An application of metaheuristic optimization algorithms for solving the flexible job-shop scheduling problem. *Oper. Res. Eng. Sci. Theory Appl.* **2020**, *3*, 13–28. [[CrossRef](#)]
17. Todorov, T.; Mitrev, R.; Penev, I. Force analysis and kinematic optimization of a fluid valve driven by shape memory alloys. *Rep. Mech. Eng.* **2020**, *1*, 61–76. [[CrossRef](#)]
18. Ganguly, S. Multi-objective distributed generation penetration planning with load model using particle swarm optimization. *Decision Making: Appl. Manag. Eng.* **2020**, *3*, 30–42. [[CrossRef](#)]
19. Negi, G.; Kumar, A.; Pant, S.; Ram, M. Optimization of Complex System Reliability using Hybrid Grey Wolf Optimizer. *Decis. Mak. Appl. Manag. Eng.* **2021**, *4*, 241–256. [[CrossRef](#)]
20. Precup, R.-E.; Preitl, S.; Petriu, E.; Bojan-Drăgăș, C.-A.; Szedlak-Stinean, A.-I.; Roman, R.-C.; Hedrea, E.-L. Model-based fuzzy control results for networked control systems. *Rep. Mech. Eng.* **2020**, *1*, 10–25. [[CrossRef](#)]
21. Gharib, M.R. Comparison of robust optimal QFT controller with TFC and MFC controller in a multi-input multi-output system. *Rep. Mech. Eng.* **2020**, *1*, 151–161. [[CrossRef](#)]
22. Belen, S.; Kropa, E.; Weber, G.-W. Rumours within time dependent Maki-Thompson model. *Preprint* **2008**, *127*. in press.
23. Goldberg, D.E.; Holland, J.H. Genetic Algorithms and Machine Learning. *Mach. Learn.* **1988**, *3*, 95–99. [[CrossRef](#)]
24. 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; IEEE: Piscataway, NJ, USA, 1995; Volume 4, pp. 1942–1948.
25. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. GSA: A gravitational search algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [[CrossRef](#)]
26. Rao, R.V.; Savsani, V.J.; Vakharia, D. Teaching–learning–based optimization: A novel method for constrained mechanical design optimization problems. *Comput.-Aided Des.* **2011**, *43*, 303–315. [[CrossRef](#)]
27. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
28. Saremi, S.; Mirjalili, S.; Lewis, A. Grasshopper optimisation algorithm: Theory and application. *Adv. Eng. Softw.* **2017**, *105*, 30–47. [[CrossRef](#)]
29. Faramarzi, A.; Heidarinejad, M.; Mirjalili, S.; Gandomi, A.H. Marine Predators Algorithm: A nature-inspired metaheuristic. *Expert Syst. Appl.* **2020**, *152*, 113377. [[CrossRef](#)]
30. Kaur, S.; Awasthi, L.K.; Sangal, A.L.; Dhiman, G. Tunicate Swarm Algorithm: A new bio-inspired based metaheuristic paradigm for global optimization. *Eng. Appl. Artif. Intell.* **2020**, *90*, 103541. [[CrossRef](#)]
31. Janečka, J.E.; Jackson, R.; Yuquang, Z.; Diqiang, L.; Munkhtsog, B.; Buckley-Beason, V.; Murphy, W. Population monitoring of snow leopards using noninvasive collection of scat samples: A pilot study. *Anim. Conserv.* **2008**, *11*, 401–411. [[CrossRef](#)]
32. Hemmer, H. *Uncia uncia*. *Mamm. Species* **1972**, *20*, 1–5. [[CrossRef](#)]
33. Christiansen, P. Canine morphology in the larger Felidae: Implications for feeding ecology. *Biol. J. Linn. Soc.* **2007**, *91*, 573–592. [[CrossRef](#)]
34. Sunquist, M.; Sunquist, F. *Wild Cats of the World*; University of Chicago Press: Chicago, IL, USA, 2017.
35. Jackson, R.M. *Home Range, Movements and Habitat Use of Snow Leopard (Uncia uncia) in Nepal*; University of London: London, UK, 1996.
36. Fox, J.L.; Chundawat, R.S. Observations of snow leopard stalking, killing, and feeding behavior. *Mammalia* **1988**, *52*, 137–140.

-
37. Veteska, J.; Kursch, M. The research on the efficiency of the methods of talent management within organizations. *New Educ. Rev.* **2018**, *52*, 28–42. [[CrossRef](#)]
 38. Veteska, J.; Kursch, M.; Svobodova, Z.; Tureckiova, M. Longitudinal co-teaching projects—scoping review. In Proceedings of the 17th International Conference on Cognition and Exploratory Learning in Digital Age, Lisbon, Portugal, 18–20 November 2020.
 39. Coufal, P.; Hubálovský, Š.; Hubálovská, M. Application of the Basic Graph Theory in Autonomous Motion of Robots. *Mathematics* **2021**, *9*, 919. [[CrossRef](#)]