TEM Journal. Volume 12, Issue 3, pages 1281-1293, ISSN 2217-8309, DOI: 10.18421/TEM123-07, August 2023.

An Overview and Comparison of Selected State-of-the-Art Algorithms Inspired by Nature

Marko Gulić [1,2], Martina Žuškin [1], Vilim Kvaternik [3]

[1] *University of Rijeka, Faculty of Maritime Studies, Studentska 2, 51000 Rijeka, Croatia,*
[2] *Center for Artificial Intelligence and Cybersecurity, University of Rijeka, 51000 Rijeka, Croatia*
[3] *ENERCON usluge Croatia d.o.o., 10000 Zagreb, Croatia*

Abstract **– Optimization is essential in various fields such as finance, transportation, energy, and health care. However, solving real optimization problems, especially nondeterministic polynomial, requires considerable computational resources. Metaheuristics provide fast and cost-effective solutions to these problems. In this paper, eight state-of-the-art nature-inspired metaheuristic algorithms that have demonstrated excellent performance are compared in detail. In addition, a novel tournament procedure has been proposed to produce a quality ranking of selected metaheuristic algorithms, which are compared based on their optimization results, even if they were not originally tested with the same set of test functions, but only partially. The selected algorithms are evaluated using thirty-two test functions, which is a representative sample size. The evaluation also showed that while one algorithm produced the best overall results, this does not mean that this algorithm is the best for solving each function. This also highlights the need for further research in metaheuristic algorithms.**

Keywords **– Optimization, nature-inspired metaheuristics, comparison of metaheuristics, optimization functions.**

DOI: 10.18421/TEM123-07
https://doi.org/10.18421/TEM123-07

Corresponding author: Marko Gulić,
University of Rijeka, Faculty of Maritime Studies, Studentska 2, 51000 Rijeka, Croatia
Center for Artificial Intelligence and Cybersecurity, University of Rijeka, 51000 Rijeka, Croatia
Email: marko.gulic@pfri.uniri.hr

Received: 07 May 2023.
Revised: 26 July 2023.
Accepted: 01 August 2023.
Published: 28 August 2023.

1. Introduction

Optimization is a fundamental concept in science and engineering that aims to find the best solution to a particular problem [1]. In finance, for example, optimization can be used to allocate assets in a way that maximizes return and minimizes risk [2].

Mathematically, optimization of problems involves finding the most favorable solution from a set of possible options [3]. They usually consist of an objective function and constraints with either discrete or continuous variables. The complexity of such a problem depends on the size of the solution search space. Real-world optimization problems are often NP (Non-deterministic Polynomial-time) problems that can be solved with a non-deterministic algorithm. However, finding the best solution can be resource intensive, making it impractical. Metaheuristics can help overcome this challenge. In [4], the authors defined metaheuristics in computer science and mathematical optimization:

In computer science and mathematical optimization, a metaheuristic is a higher-level procedure or heuristic that aims to find, generate, or select a heuristic (partial search algorithm) that can provide a sufficiently good solution to an optimization problem.

Metaheuristics provide fast and efficient solutions, but cannot ensure global optimality unlike numerical optimization algorithms. These algorithms are popular for their stochastic optimization techniques and ease of implementation [5], and involve two phases: exploration and exploitation. However, the use of random variables means that optimal performance on one problem does not guarantee similar results on another, due to the No Free Lunch (NFL) theorem [6]. Therefore, new metaheuristic algorithms are regularly proposed in scientific literature.

Nature-inspired metaheuristic algorithms are popular for modeling natural behaviors and optimizing objective functions.

They originated in the genetic algorithm proposed in 1975 [7]. Many new algorithms have been proposed, and new algorithms are constantly emerging due to the NFL theorem. Section 2 reviews the numerous nature-inspired metaheuristic algorithms in the literature.

Moreover, a detailed survey and mutual comparison of eight selected modern nature-inspired metaheuristic algorithms is performed in this paper. These are the Blue Monkey (BM) [8], Bear Smell Search Algorithm (BSSA) [9], Deer Hunting Optimization Algorithm (DHOA) [10], Golden Eagle Optimizer Algorithm (GEO) [11], Squirrel Search Algorithm (SSA) [12], Aquila Optimizer (AO) [13], Pelican Optimization Algorithm (POA) [14], and Snow Leopard Optimization Algorithm (SLOA) [15]. These algorithms were chosen for cross-comparison because of their significant results in finding the optimal solution for the test functions used in the works that propose the above algorithms. In addition, these algorithms were chosen because they are among the most recent metaheuristics proposed in the literature. Moreover, the algorithms have a very clear, simple, and well-explained process for finding optimal solutions to a problem. Section 3 provides a more detailed description of these algorithms.

In the literature, algorithms are often compared not to modern nature-inspired metaheuristic algorithms, but to well-known algorithms (e.g., GA), so it is sometimes difficult to determine which of the modern algorithms are the best. The next problem that arises when comparing modern nature-inspired metaheuristic algorithms is the lack of standardization in testing the proposed algorithms on a standardized set of optimization functions. Although the scientific field of nature-inspired metaheuristic algorithms is already well researched, authors still independently determine on which test functions their algorithm is evaluated. For this reason, this paper proposes a new method for comparing algorithms that have not been tested on the same set of functions without requiring all of these algorithms to be implemented independently and tested locally. Therefore, a tournament method principle for comparing algorithms has been proposed. In this method, a pairwise comparison is made between each algorithm and each other algorithm individually. The values within the results of the common functions are used to determine which of the two compared algorithms won the duel. The winning algorithm is assigned points. In the end, the sum of all the points is calculated, and the algorithm with the most points is indeed the best algorithm among the selected algorithms. Furthermore, this evaluation method allows the comparison of any two algorithms or a set of algorithms to determine the most optimal one.

This evaluation involved the utilization of thirty-two test functions, on which a minimum of two out of the eight selected algorithms were tested. A detailed process of evaluation is described in Section 4.

The paper is structured as follows. In Section 2, an overview is presented on the multitude of metaheuristic algorithms that are inspired by nature and have been documented in the literature until now. In section 3, the metaheuristic algorithms involved in the evaluation in the context of this paper (BM, BSSA, DHOA, GEO, SSA, AO, POA and SLOA) are presented in detail. Section 4 deals with the evaluation of the selected metaheuristics and presents the results of the performed evaluation. Section 5 contains the conclusion.

2. Nature Inspired Metaheuristics in Literature

There are numerous metaheuristic algorithms inspired by nature, including modified and hybrid algorithms, in addition to those compared in this paper. Thus, here we briefly review the original nature-inspired metaheuristic algorithms and their inspiration by natural behavior. These algorithms mimic the phases of exploration and exploitation using nature's behavior.

The genetic algorithm (GA) is the most popular metaheuristic inspired by nature [7]. It uses natural selection to reproduce the best solutions and mimics the mutation process for new solutions. Simulated Annealing (SA) [16] draws inspiration from the annealing process observed in metallurgy and reduces randomness while accepting suboptimal solutions with a certain probability to find the global optimum. Particle Swarm Optimization (PSO) [17] derives inspiration from collective behavior observed in social organisms and moves particles within the search space using mathematical formulas to guide them to optimal solutions.

Between 2000 and 2010, significant research was conducted on behaviors from nature that can be used and modeled in metaheuristic algorithms to solve optimization problems. Among the best-known algorithms developed during this period are the Firefly algorithm (FF) [18], Ant Colony Optimization (ACO) [19], the Artificial Bee Colony (ABC) [20], and the Bat algorithm (BA) [21]. The ABC mimics honeybee foraging behavior and implements two behaviors critical to self-organization and collective intelligence. ACO draws inspiration from the behavior of real ants, which communicate through pheromones to guide each other to resources. BA is inspired by foraging behavior of microbats, using artificial microbats as search agents to determine the best prey (solution).

FF is inspired by the blinking behavior of fireflies, and solutions are treated like fireflies, with their brightness depending on how well they perform the objective function. Other well-known algorithms from this period include the Artificial Fish Swarm Algorithm (AFSA) [22] and Monkey Search (MS) [23]. AFSA draws inspiration from the social behavior of fish, while MS imitates the search behavior of monkeys to search for optimal solutions.

A variety of nature-inspired metaheuristic algorithms were introduced in the period from 2010 to 2020. The Flower Pollination Algorithm (FPA) imitates the reproductive process in flowering plants [24]. The Fruit Fly Optimization Algorithm (FFOA) [25] takes inspiration from how fruit flies search for food. Krill Herd (KH) [26] is a metaheuristic that draws inspiration from the herding behavior of individual krill organisms and searches for optimal solutions in a multidimensional space by locating areas of high herd and food density. The Dolphin Echo-Location (DE) algorithm [27] is a method inspired by the hunting strategies of dolphins. Furthermore, in [28], the authors propose the Gray Wolf Optimizer (GWO), which simulates a pack of wolves searching for prey. The next algorithm is the Lightning Search Algorithm (LSA) [29], which draws inspiration from the lightning propagation. The life behavior of microalgae is described in [30], where the artificial algae algorithm (AAA) is proposed. The algorithm AAA models the behavior of algae, including movement toward light for photosynthesis, adaptation to the environment, change of dominant species, and reproduction by mitotic division. Furthermore, the Ant Lion Optimizer (ALO) has been proposed in the literature, inspired by the interaction of ants and ant lions in nature [31]. Ant lions are predatory insects that feed on ants in the larval stage. In 2016, three other metaheuristics inspired by fish behavior were proposed: the Shark Smell Optimization (SSO) [32], which mimics the olfactory system of sharks, the Dolphin swarm optimization algorithm (DSOA) [33], which mimics the strategy of dolphins hunting sardine schools, and the Whale Optimization Algorithm (WOA) [34], which simulates the hunting strategy of these marine mammals. In addition, two other interesting algorithms based on deep-sea behavior have been proposed, the Salp Swarm Algorithm [35] and the Marine Predators Algorithm (MPA) [36]. SSA draws inspiration from the navigational ability and behavior of salps while searching for food, while MPA is a metaheuristic inspired by nature guided by the principles of optimal foraging and predator-prey encounter rate policy in marine ecosystems.

Furthermore, the behavior of virus propagation in nature has also led to the development of metaheuristics. Virus Colony Search (VCS) [37] and a metaheuristic mimicking the Ebola virus [38] have been proposed to mimic the propagation and infection techniques of viruses in a cellular environment. Another interesting algorithm is the Crow Search Algorithm (CSA) [39]. It mimics the habits of these birds, which are able to store food in multiple locations and retrieve it when needed. The Lion Optimization Algorithm (LOA) [40] aims to mimic the social and hunting strategies of lions by incorporating features such as a hierarchical structure and cooperative search behavior that enable local and global search strategies. In addition to LOA, which simulates the behavior of jungle animals, another algorithm called Spotted Hyena Optimizer (SHO) was proposed in [41]. SHO draws inspiration from the social behavior and cooperation strategies of these interesting animals. The key concept of SHO is to simulate group interactions between these animals. Moreover, the Grasshopper Optimization Algorithm (GOA) [42] and the Butterfly-Inspired Optimization Algorithm (BFOA) [43] from 2017 are also interesting. GOA was inspired by the swarming and feeding behavior of grasshoppers, while BFOA draws inspiration from the collective behavior of butterflies, which involves independent exploration and social communication through pheromones to find food sources. In addition to SLOA, POA, AO, GEO and BSSA, which will be part of the analysis and mutual comparison in this paper, as well as the Ebola and MPA algorithms mentioned earlier, the Tunicate Swarm Algorithm (TSA) [44] and Sand Cat Swarm Optimization (SCSO) [45] are also mentioned here as representatives of the period from 2020 to the present. TSA was inspired by the coordinated feeding behavior of tunicates through the release of pheromones, similar to the feeding behavior of ants and bees. Considering the inspiration of the SCSO, it mimics the hunting behavior of these animals. The sand cats locate and attack their prey using sound frequencies. The goal of each sand cat is to capture higher value prey.

Numerous studies have addressed the development of novel metaheuristic algorithms inspired by natural behavior. Despite the growing interest in recent years, it is difficult to compare these algorithms because they are often tested on different functions. To address this problem, we propose a new model for comparing multiple algorithms that is not limited to nature-inspired metaheuristic algorithms. Due to their popularity, research in this area is expected to continue.

3. Overview of Selected State-of-the-Art Algorithms Inspired by Nature

The following chapter presents an overview of carefully selected state-of-the-art nature-inspired algorithms and their respective pseudocodes, which are evaluated in Section 4. As mentioned earlier, these algorithms were selected based on their good results achieved in optimizing a large number of test optimization functions. Moreover, the process of executing these algorithms is very simple and efficient. In addition, these algorithms are among the most recent metaheuristics proposed in the literature. In the following, the steps of each of these algorithms in finding solutions to optimization problems are explained in detail.

Blue monkey. Blue monkey (BM) [8] algorithm draws inspiration from the behavior of blue monkeys (Cercopithecus mitis) within the groups in which they live. They live in larger groups dominated by one male, who dominates the entire group and determines the movement of the group in search of a food source.

Each young male leaves the group very early to find his new group [46]. The young male challenges the dominant male of another group. If he defeats him, he becomes the new leader of this group. The blue monkey algorithm is based precisely on the movement of a single group as well as on the phenomenon of group takeover by young males. In this algorithm, mature blue monkeys (individuals representing the solution) are divided into groups. There is also an additional group of young blue monkeys. As the algorithm is executed iteratively, the strongest young monkeys (individuals) enter the forming groups of mature monkeys and displace them from the groups. The stronger a monkey is (the better the result of this individual), the stronger is its position in the group hierarchy, and in this way the weaker monkeys follow the stronger ones during the algorithm execution process. Therefore, the position of each monkey (solution) within a group is determined by the position of the best monkey (solution) within that group. The updating of the positions of the monkeys is done at each iteration. Algorithm 1 shows the pseudocode of BM.

```
Procedure blueMonkey
  Data:  N – the size of population;
         MAX – maximum number of iterations;
  Result: X – the best solution within the population P
  Initialize population of blue monkeys and
          children monkeys, bi (i = 1 to N);
  Initialize Power Rate (Rate) and Weight (W),
          with Rate in range [0, 1] and W in range [4, 6];
  Distribute blue monkeys randomly into teams (T),
          with all children in one team;
  Calculate fitness of children and blue monkeys in each group;
  for each group:
  |  a. Select best and worst blue monkeys according to fitness
  |        values, store as Current Best;
  |  b. Children select best fitness;
  t = 1;
  while (t ≤ MAX)
  |  Swapping the worst fitness in each group by the
  |                      best fitness in children group;
  |  Update Rate and X position of all blue monkeys in each
  |                      group by Equations 1 and 2 [8];
  |  Update Rate and X position of children by Equations 3 and 4 [8];
  |  Update the fitness of all blue monkey and children;
  |  if (New Best is better than Current Best)
  |   | Current Best=New Best;
     t = t + 1;
  X ⟵  select the best blue monkey (solution)
```

Algorithm 1. Pseudocode of BM algorithm [8]

Bear Smell Search Algorithm. The BSSA is a nature-inspired algorithm that mimics the dynamic behavior of bears, which have an extraordinary sense of smell and use it to find food over long distances [9]. Bears have large olfactory bulbs that allow them to detect different smells, making their sense of smell incredibly powerful. This is because the olfactory bulb is a neural model of the vertebrate forebrain that allows for strong intensification and diversification in optimization. The algorithm is based on the bear's movement patterns, which are determined by the value of different odors [47], [48].

Each odor represents a solution of the bear smell algorithm and consists of a set of components (variables) of an overall solution. The process is repeated at each iteration until the condition is met. Algorithm 2 shows the pseudocode of BSSA.

Deer Hunting Optimization Algorithm. The DHOA is a metaheuristic algorithm inspired by the hunting tactics of hunters in pursuit of deer [10].

The hunting method relies on the synchronized movement of two hunters moving towards the best positions until they reach the target deer.

The goal of the proposed metaheuristic is to determine the best hunting position for humans, which requires studying the behavior and characteristics of deer. Deer has five times better vision than humans, can see 250 to 270 degrees in the periphery, and can perceive even the smallest movements.

```
Procedure bearSmellSearchAlgorithm
  Data:  N – the size of population; LB – lower bounds;
         D – number of variables; UB – upper bounds;
         MAX – maximum number of iterations;
  Result: X – the best solution within the population P
  Generate the initial random population with dimension
                                  OM = N×D in search space;
  Evaluate fitness for each row (solution) of matrix OM;
  Save best odor (solution) as current global solution, O^g;
  for iter = 1 : MAX
  |  Calculate the maximum output of the glomerular activity
  |        using equation 2; MG;
  |  Obtain the DS based on the breathing function in equation 1;
  |  Calculate the cell output functions for the mitral and
  |        granular cells, f_x(x) and f_y(y), using equations 4-5;
  |  Define H_0, W_0, and L_0 based on equations 6-8;
  |  Solve equation 3;
  |  Set vectors POC, POF, and OF based on equations 9-10;
  |  Calculate vectors EOF and DOC using equations 11-12;
  |  Set two thresholds ζ_1 and ζ_2;
  |  Calculate coefficients C_1 to C_4;
  |  for i = 1 : N
  |  |  if (criterion is met)
  |  |  |  Generate a new population based on the first
  |  |  |           part of equation 13;
  |  |  else
  |  |  |  Generate a new population based on the second
  |  |  |           part of equation 13;
  X ⟵  select the best odor (solution)
```

Algorithm 2. Pseudocode of BSSA algorithm [9]

In addition, deer have a keen sense of alertness and can detect danger by smelling 60 times better than humans and alerting others by kicking violently and sniffing loudly. Deer also have the unique ability to perceive ultra-high frequency sounds that humans cannot. Therefore, the movements of hunters must be incorporated into the algorithm, as well as the behavior and characteristics of the deer themselves. The pseudocode of DHOA is presented in Algorithm 3.

Golden Eagle. The Golden Eagle Optimizer (GEO) algorithm draws inspiration from the ability of golden eagles to adjust their speed as they move through distinct phases of their spiraling hunting trajectory [11].

```
Procedure deerHuntingOptimizationAlgorithm
  Data:  N – the size of population Y;
         MAX – maximum number of iterations;
  Result: X – the best solution within the population Y
  Generate the initial random population Y on N solutions;
  for t = 1 : MAX
  |  for i = 1 : N
  |  |  Compute the fitness of solution i;
  |  |  Update a, d, A, p, X, L and b;
  |  |  if (p < 1)
  |  |  |  if (|L| ≥ 1)
  |  |  |  |  Update the position of the individual
  |  |  |  |           using equation 4;
  |  |  |  else
  |  |  |  |  Update the position of the individual
  |  |  |  |           using equation 12;
  |  |  else
  |  |  |  Update the position of the individual
  |  |  |           using equation 11;
  |  Compute the fitness of each solution;
  |  Update Y^lead and Y^successor;
  X ⟵  select the best solution (Y^lead)
```

Algorithm 3. Pseudocode of DHOA algorithm [10]

The hunting process of golden eagles has several defining features, including a spiral search trajectory and straight attack path, a gradual transition from cruising to attacking, a simultaneous ability to both cruise and attack, and seeking in-formation from other eagles about prey. The interplay between cruising and attacking is an example of exploration and exploitation that can be modelled mathematically for devising a metaheuristic algorithm. At each iteration, an eagle randomly selects the target of another eagle and circles the best location it has found so far. An eagle also has the option to orbit its own stored location. Within the algorithm, the prey represents the best solution determined by one of the positions of all golden eagles. At each iteration, golden eagles select prey from the swarm's memory and determine their attack and cruise vectors based on the selected prey. Algorithm 4 shows the pseudocode of GEO.

Squirrel Search Algorithm. The Squirrel Search Algorithm (SSA) is inspired by foraging of flying squirrels [12]. When weather conditions are favorable, flying squirrels glide among the trees in search of food. They consume acorns when available, and then search for optimal food sources, such as hickory nuts, for winter storage.

```
Procedure goldenEagleOptimizer
  Data: N – the size of population P;
        MAX – maximum number of iterations;
  Result: X – the best solution within the population P
  Generate the initial random population P on N solutions;
  Evaluate fitness function for each solution i in P;
  Initialize population memory;
  Initialize pa and pc;
  for t = 1 : MAX
  |  Update pa and pc with Equation 9;
  |  for i = 1 : N
  |  |  Randomly select a prey from the population's memory;
  |  |  Calculate attack vector A with Equation 1;
  |  |  if (attack vector's length ≠ zero)
  |  |  |  Calculate cruise vector C with Equations 2-5;
  |  |  |  Calculate step vector Δx with Equations 6, 7, and 8;
  |  |  |  Update position with Equation 8;
  |  |  |  Evaluate fitness function for the new position;
  |  |  |  if (fitness is better than the fitness of the position
  |  |  |              in eagle i's memory)
  |  |  |  |  Replace the new position with the position in
  |  |  |  |        eagle i's memory;
  X ⟵  select the best prey (solution)
```

Algorithm 4. Pseudocode of GEO algorithm [11]

During winter, they become less active but do not hibernate and face increased risk of predation due to the loss of leaf cover. This cycle repeats throughout their lifespan and forms the basis of SSA. The peculiarity of this algorithm lies in the use of the Lévy distribution, a powerful mathematical tool used to improve global exploration not only in this squirrel algorithm but also in other metaheuristic algorithms [49], [50], [51]. Algorithm 5 shows the pseudocode of SSA algorithm.

Aquila Optimizer. The Aquila Optimizer (AO) metaheuristic draws the inspiration from the aquila's hunting behaviors in nature [13]. The aquila is a skilled hunter known for its speed and claws, and it is often studied. Male aquila catch more prey when hunting alone. The bird uses four different hunting methods and can quickly switch between them.

To catch birds in flight, aquila firstly uses hunting strategy of high soar with a vertical stoop. This involves a high flight followed by a low angle glide and a sudden dive with wings and tail spread to seize prey. In addition, aquila often uses contour flight with short glides (the second method of attack), flying at low altitudes to pursue prey on the ground or in the air. This technique is useful for hunting a variety of animals, including seabirds and ground squirrels. The third hunting method of aquila is low flight with slow descent, gradually approaching prey before landing on its neck and back to attack it and used to hunt slow-moving prey with poor escape response. Finally, the fourth hunting method of aquila is to run on the ground to pull the young of large prey from their hiding places. Algorithm 6 shows the pseudocode of AO.

```
Procedure squirrelSearchAlgorithm
  Data: N – the size of population P;
  Result: X – the best solution within the population P
  Generate the initial random population P on N flying
        squirrels using Equation 2;
  Evaluate fitness of each flying squirrel's location (solution);
  Sort the locations of flying squirrels in ascending order
        depending upon their fitness value;
  Declare the flying squirrels on hickory nut tree, acorn
        nuts trees and normal trees;
  Randomly select some flying squirrels which are on
        normal trees to move towards hickory nut tree and the
        remaining will move towards acorn nuts trees
  while (the stopping criterion is not satisfied)
  |  for t = 1 : n1 (n1 = total flying squirrels which are on
  |  |     acorn trees and moving towards hickory nut tree)
  |  |  if (R1 ≥ Pdp)
  |  |  |  resolve Equation 4. a);
  |  |  else
  |  |  |  resolve Equation 4. b);
  |  for t = 1 : n2 (n2 = total flying squirrels which are on
  |  |     normal trees and moving towards acorn trees)
  |  |  if (R2 ≥ Pdp)
  |  |  |  resolve Equation 5. a);
  |  |  else
  |  |  |  resolve Equation 5. b);
  |  for t = 1 : n3 (n3 = total flying squirrels which are on
  |  |     normal trees and moving towards hickory nut tree)
  |  |  if (R3 ≥ Pdp)
  |  |  |  resolve Equation 6. a);
  |  |  else
  |  |  |  resolve Equation 6. b);
  |  Calculate seasonal constant (Sc) using Equation 12;
  |  if (Seasonal monitoring condition is satisfied)
  |  |  Randomly relocate flying squirrels using Equation 14;
  |  Update the minimum value of seasonal constant (Smin)
  |        using Equation 13;
  X ⟵  The location of squirrel on hickory nut tree is the
        final optimal solution
```

Algorithm 5. Pseudocode of SSA algorithm [12]

Pelican Optimization Algorithm. The POA is a new metaheuristic based on the behavior of pelicans when searching for prey [14]. Pelicans often work together when hunting. After spotting the location of their prey, they swoop down on the target from a height of about 15 meters, although some species prefer to descend from lower heights. Then the pelicans stretch their wings as they glide over the surface of the water to propel the fish toward shallow areas where they can be caught effortlessly. While catching the prey, a considerable amount of water enters the bird's beak, causing the bird to push its head forward to remove the excess water before swallowing the prey. Algorithm 7 shows the pseudocode of POA.

Snow Leopard Optimization Algorithm. The SLOA metaheuristic draws inspiration from the natural behaviors of snow leopards [15]. Snow leopards use scent signs and move in zigzags to indicate their location and to track each other. They also hunt by using rocky cliffs for cover and approaching their prey. Once they reach the proper distance from the prey, they slowly walk a short distance, then suddenly run, and bite the prey in the neck to kill it. Leopard behavior is modelled by four components, which include the migration routes and search for prey briefly discussed above, as well as reproduction and mortality. Algorithm 8 shows the pseudocode of SLOA.

```
Procedure aquilaOptimizer
  Data:  N – the size of population X;
  Result: X – the best solution within the
        population P
  Initialize the population X of the AO using
        Equation 47;
  Initialize the parameters of the AO (i.e., α, δ, etc);
  while (the stopping criterion is not satisfied)
      Evaluate fitness function of each solution;
      Xbest(t)= Determine the best obtained solution
            according to the fitness values;
      for i = 1 : N
          Update the mean value of the current
              solution XM(t);
          if (t ≤ (2/3) * T)
              if (rand ≤ 0.5)
                  {Step 1: Expanded exploration (X1)}
                  Update the current solution using
                        Equation (3);
                  if (Fitness(X1(t+1)) < Fitness(X(t)))
                      X(t) =(X1(t+1));
                      if (Fitness(X1(t+1)) < Fitness(Xbest(t)))
                          Xbest (t) =(X1(t+1));
              else
                  {Step 2: Narrowed exploration (X2)}
                  Update the current solution using
                        Equation (5);
                  if (Fitness(X2(t+1)) < Fitness(X(t)))
                      X(t) =(X2(t+1));
                      if (Fitness(X2(t+1)) < Fitness(Xbest(t)))
                          Xbest (t) =(X2(t+1));
          else
              if (rand ≤ 0.5)
                  {Step 3: Expanded exploitation (X3)}
                  Update the current solution using
                        Equation (13);
                  if (Fitness(X3(t+1)) < Fitness(X(t)))
                      X(t) =(X3(t+1));
                      if (Fitness(X3(t+1)) < Fitness(Xbest(t)))
                          Xbest (t) =(X3(t+1));
              else
                  {Step 4: Narrowed exploitation (X4)}
                  Update the current solution using
                        Equation (14);
                  if (Fitness(X4(t+1)) < Fitness(X(t)))
                      X(t) =(X4(t+1));
                      if (Fitness(X4(t+1)) < Fitness(Xbest(t)))
                          Xbest (t) =(X4(t+1));
  X ← the best solution Xbest
```

Algorithm 6. Pseudocode of AO algorithm [13]

4. Evaluation Results and Discussion

In this section, optimization results for selected nature-inspired metaheuristics, which are described in detail in Section 3, are presented. As stated before, the reason for selecting these algorithms for cross-comparison is their significant performance in finding optimal solutions for the test functions used in this work. Furthermore, these algorithms are among the state-of-the-art nature-inspired metaheuristics recently published. Moreover, their procedure for finding optimal solutions to a problem is well explained, simple, and straightforward. For this evaluation, thirty-two objective test functions have been used to compare selected state-of-the-art nature inspired metaheuristics. Due to the limited length of the paper, we have not provided all the details of these test objective functions (just names), which can be found in papers that describes the algorithms GEO, BM, AO, BSSA, DHOA, POA, SLOA, and SSA.

```
Procedure pelicanOptimizationAlgorithm
  Data:  N – the size of population X
         T – number of iterations
  Result: X – the best solution within the population X
  Initialization of the position of pelicans (Equation 52)
         in population X;
  Calculate the objective function for each pelican;
  for t = 1 : T
  |  Generate the position of the prey at random;
  |  for i = 1 : N
  |  |  Phase 1: Moving towards prey (exploration phase)
  |  |  for j = 1 : m
  |  |  |  Calculate new status of the j-th dimension
  |  |  |        using Equation 4;
  |  |  Update the i-th population member using
  |  |        Equation 5;
  |  |  Phase 2: Winging on the water surface
  |  |        (exploitation phase)
  |  |  for j = 1 : m
  |  |  |  Calculate new status of the j-th dimension
  |  |  |        using Equation 6;
  |  |  Update the i-th population member using
  |  |        Equation 7;
  |  Update best candidate solution
  X⟵  the best candidate solution obtained by POA
```

Algorithm 7. Pseudocode of POA algorithm [14]

The optimization results for these test functions obtained with the selected metaheuristics GEO, BM, AO, BSSA, DHOA, POA, SLOA, and SSA are shown in Table 1. It is important to note that none of the evaluated algorithms has results available for all thirty-two selected functions.

Although all the selected functions are well-known functions for testing metaheuristic algorithms, in papers where new metaheuristics are proposed, the authors do not always select the same set of functions for testing. Thus, the problem arises of how to compare multiple algorithms when they are not tested with the same set of functions. These thirty-two functions were selected in such a way that optimization results were extracted from the papers describing the selected algorithms for those functions for which at least two of the eight selected algorithms were tested. To be able to determine which of the selected algorithms is generally the best, it is necessary to develop a method that would determine the best algorithm in a meaningful way despite the different functions tested.

```
Procedure snowLeopardOptimizationAlgorithm
  Data:  N – the size of population X
         T – number of iterations
  Result: X – the best solution within the population X
  Generate an initial snow leopard population matrix;
  Calculate the objective function for each snow leopard;
  for t = 1 : T
  |  Phase 1: Travel routes and movement
  |  for i = 1 : N
  |  |  for d = 1 : m
  |  |  |  Calculate x_{i,d}^{P1} using Equation 3;
  |  |  Update X_i using Equation 4;
  |  Phase 2: Hunting
  |  for i = 1 : N
  |  |  for d = 1 : m
  |  |  |  Calculate location of prey p_{i,d};
  |  |  |  Calculate x_{i,d}^{P2} using Equation 7;
  |  |  Update X_i using Equation 8;
  |  Phase 3: Reproduction
  |  for l = 1 : 0.5 x N
  |  |  Generate cub Cl using Equation (9);
  |  Phase 4: Mortality
  |  Adjust the number of snow leopards to N due
  |   to mortality based on criterion of the objective function;
  |  Save best solution obtained with the SLOA so far;
  X⟵  the best candidate solution obtained by SLOA
```

Algorithm 8. Pseudocode of SLOA algorithm [15]

In this paper, we conducted a comparative analysis of eight algorithms using a league tournament approach. We compared algorithms based on the common functions for which this pair of algorithms has optimization results. The algorithm, within the compared pair of algorithms, that obtained better optimization results for a larger number of functions, won the duel. Table 2 shows an example of a duel between the algorithms GEO and SSA.

Table 1. Results of the optimization of thirty-two test functions with the algorithms GEO, BM, AO, BSSA, DHOA, POA, SLOA, SSA

Test functions	GEO	BM	AO	BSSA	DHOA	POA	SLOA	SSA	F_{min}
F1 - Ackley	1,98E-01	1,06	8,88E-16	8,87E-18	0	8,88E-16	4,44E-15	1,39E-04	0
F2 - Beale	0	N/A	N/A	0	2,44E-08	N/A	N/A	9,56E-18	0
F3 - Bohachevsky 1	N/A	-1,03	N/A	0	N/A	N/A	N/A	0	0
F4 - Booth	N/A	2,36E-31	N/A	0	0	N/A	N/A	9,59E-25	0
F5 - Exponential	-1	-2	N/A	N/A	N/A	N/A	N/A	N/A	-1
F6 - Colville	N/A	N/A	N/A	5,43E-12	N/A	N/A	N/A	1,43E-09	0
F7 - Dixon-Price	N/A	-195	N/A	6,54E-04	N/A	N/A	N/A	2,24E-01	0
F8 - Drop wave	-1	N/A	N/A	-1	N/A	N/A	N/A	N/A	-1
F9 - Easom	N/A	-3,86	N/A	-1	-0,46	N/A	N/A	-1	-1
F10 - Egg holder	-928	3,98	N/A	-959,6	-4302	N/A	N/A	N/A	-959,64
F11 - Goldstein Price	N/A	-1	3	3	1,83	3	3	N/A	3
F12 - Griewank	5,01E-03	N/A	0	0	N/A	0	0	3,44E-06	0
F13 - Hartmann 3-D	N/A	1,00E-89	-3,8624	-3,862	-0,18	-3,8628	-3,8628	N/A	-3,8628
F14 - Hartmann 6-D	N/A	0	-3,3014	N/A	N/A	-3,322	-3,322	N/A	-3,3224
F15 - Levy N. 13	1,35E-31	N/A	N/A	3,76E-08	8,23E-32	N/A	N/A	N/A	0
F16 - Matyas	1,99E-94	-1	N/A	0	0	N/A	N/A	1,54E-22	0
F17 - Penalized 1	2,08E-02	1,56E-03	2,93E-05	N/A	N/A	2,73E-16	1,06E-02	N/A	0
F18 - Penalized 2	7,93E-03	0	2,23E-02	N/A	N/A	2,83E-15	0,11	N/A	0
F19 - Powell	N/A	7,45E-01	N/A	0	50,66	N/A	N/A	N/A	0
F20 - Quatric	N/A	0	8,29E-04	2,56E-07	N/A	9,37E-06	1,47E-04	5,02E-01	0
F21 - Rastrigin	1,01E+01	N/A	0	0	0	0	0	4,91E-07	0
F22 - Rastrigin	4,17	N/A	1,89E-03	4,73E-01	102	27	24,26	9,45E-01	0
F23 - Schaffer	N/A	-1,8	N/A	9,74E-06	0	N/A	N/A	9,72E-04	0
F24 - Schubert	N/A	N/A	N/A	-182,3	N/A	N/A	N/A	-186,73	-186,73
F25 - Schwefel 1.2	N/A	8,88E-16	0	N/A	N/A	1,88E-266	1,86E-56	1,69E-05	0
F26 - Schwefel 2.21	N/A	33,68	9,91E-218	N/A	N/A	2,37E-133	8,80E-152	N/A	0
F27 - Schwefel 2.22	N/A	3,61E-41	9,50E-218	N/A	N/A	1,43E-128	2,70E-288	5,19E-04	0
F28 - Six-Hump Camel	N/A	9,86E-95	-1,03	-1,03	N/A	-1,03	-1,03	-1,03	-1,03
F29 - Sphere	4,56E-12	9,04E-11	0	0	N/A	2,87E-258	0	4,17E-08	0
F30 - Step	3,22E-14	N/A	N/A	N/A	N/A	N/A	N/A	0	0
F31 - Three-Hump Camel	6,28E-126	7,10E-05	N/A	0	N/A	N/A	N/A	N/A	0
F32 - Zakharov	N/A	1,35E-02	N/A	0	N/A	N/A	N/A	5,22E-09	0

It can be seen that the common functions for the GEO and SSA algorithms are F1, F2, F12, F16, F21, F22, F29, and F30. The SSA algorithm achieved a better optimization result for 5 functions: F1, F12, F21, F22 and F30. The GEO algorithm achieved a better optimization result for 3 functions: F2, F16, F29. So, the result is 5:3 for SSA and this algorithm won the duel.

Table 2. An example of a duel between the algorithms GEO and SSA

Test functions	GEO	SSA	F_{min}	|GEO-OS|	|SSA-OS|
F1	1,98E-01	1,39E-04	0	0,198	**0,000139**
F2	0	9,56E-18	0	**0**	9,5584E-18
F12	5,01E-03	3,44E-06	0	**0,00501**	0,00000344
F16	1,99E-94	1,54E-22	0	**1,99E-94**	1,542E-22
F21	1,01E+01	4,91E-07	0	10,1	**0,000000491**
F22	4,17	9,45E-01	0	4,17	**0,945**
F29	4,56E-12	4,17E-08	0	**4,56E-12**	4,17E-08
F30	3,22E-14	0	0	3,22E-14	**0**
				Final:	**GEO 3 : 5 SSA**

The overall results of the algorithm pair comparisons and their performance on the tested functions are shown in Table 3. In addition, the algorithm that won the duel receives 2 points, which are used for the final ranking of the algorithms. In cases where both algorithms within a pair achieve better solutions for an equal number of functions, they each receive one point.

Table 3. The overall results of the algorithm pair comparisons and their performance on the tested functions

Algorithms	BSSA	AO	POA	SLOA	DHOA	SSA	GEO	BM
BSSA		2:2	4:1	3:1	7:3	13:1	8:1	15:1
AO	2:2		5:5	6:4	3:1	8:0	6:1	10:2
POA	1:4	5:5		5:4	3:1	7:1	6:1	10:2
SLOA	1:3	4:6	4:5		3:1	7:1	5:3	9:3
DHOA	3:7	1:3	1:3	1:3		5:3	4:3	7:2
SSA	1:13	0:8	1:7	1:7	3:5		5:3	8:5
GEO	1:8	1:6	1:6	3:5	3:4	3:5		6:2
BM	1:15	2:10	2:10	3:9	2:7	5:8	2:6	

The final ranking of the algorithms can be seen in Table 4. BSSA proved to be the best algorithm, achieving 6 wins and one draw in the direct duel with the other algorithms in the optimization of the given functions.

It should be emphasized that while BSSA was the best algorithm, it performed worse on some functions than the other nature-inspired metaheuristics (difference 52:12). This result only confirms the NFL theorem mentioned earlier, which states that there is no guarantee that the superior performance of an optimization algorithm on a given set of problems will lead to similar performance on other problems. This is further evidenced by the fact that in a head-to-head duel with the BM algorithm, which performed worst in this evaluation, BSSA won 15:1, meaning that BM performed better for one of the 16 test functions for which they were compared. Furthermore, the NFL theorem confirms that future research into new nature-inspired metaheuristic algorithms is warranted.

Table 4. The final ranking of the assessed algorithms

Rank	Algorithm	#comparisons	Wins	Losses	Tied	F difference	Points
1.	**BSSA**	7	6	0	1	52 : 12	13
2.	**AO**	7	5	0	2	40 : 15	12
3.	**POA**	7	5	1	1	37 : 18	11
4.	**SLOA**	7	4	3	0	31 : 22	8
5.	**DHOA**	7	3	4	0	22 : 24	6
6.	**SSA**	7	2	5	0	19 : 48	4
7.	**GEO**	7	1	6	0	17 : 36	2
8.	**BM**	7	0	7	0	17 : 65	0

After this evaluation, it is also clear that it would be useful to create a test set of functions against which, for example, any new nature-inspired metaheuristic algorithm presented in the future would be mandatory to evaluate. In this way, a standardization of the evaluation of these types of algorithms would be achieved. If this is not possible, the evaluation results of the selected algorithms can always be compared in the way proposed in this paper.

5. Conclusion

The concept of optimization plays a central role in science and engineering, as it aims to find the optimal solution to a given problem. Many real-world optimization tasks fall into the category NP, which makes it challenging to obtain the optimal solution because they often require a significant number of resources, making them impractical or ineffective. Nature-inspired metaheuristics are powerful optimization algorithms that have demonstrated remarkable performance in finding optimal or suboptimal solutions to a wide range of problems, including NP.

With an overview of nature-inspired metaheuristic algorithms from early research to the present, this paper also provides an overview and evaluation of eight state-of-the-art nature-inspired metaheuristics, GEO, BM, AO, BSSA, DHOA, POA, SLOA, and SSA, using thirty-two objective test functions.

The evaluation was performed in a kind of league tournament to compare each algorithm with each algorithm based on the common functions for which this pair of algorithms achieves optimization results. The algorithm that obtained better optimization results for a larger number of functions won the duel, and the final ranking of the algorithms showed that BSSA was the best algorithm. Notably, it should be emphasized that although BSSA was the best algorithm, it achieved worse results than the other algorithms for some functions. This result confirms the NFL theorem, which states that there is no assurance that the superior performance of an optimization algorithm on a given set of problems will lead to similar performance on other problems.

To summarize, this work has provided important insights into the performance of nature-inspired metaheuristics and demonstrated the importance of evaluating new algorithms against a standard set of test functions. This approach can help researchers compare and select the best algorithm for a given optimization problem. Overall, nature-inspired metaheuristics continue to be actively explored in scientific research, and the results of this study will contribute to future research in this area.

References:

[1]. Papadimitriou, C. H., & Steiglitz, K. (1998) *Combinatorial Optimization: Algorithms and Complexity*. Courier Corporation.

[2]. Markowitz, H. (1952). PORTFOLIO SELECTION*. *The Journal of Finance*, *7*(1), 77–91. Doi: 10.1111/j.1540-6261.1952.tb01525.x

[3]. Hajipour, V., Kheirkhah, A., Tavana, M., & Absi, N. (2015). Novel Pareto-Based Meta-Heuristics for Solving Multi-Objective Multi-Item Capacitated Lot-Sizing Problems. *The International Journal of Advanced Manufacturing Technology*, *80*, 31–45. Doi: 10.1007/s00170-015-6993-6

[4]. Bianchi, L., Dorigo, M., Gambardella, L. M., & Gutjahr, W. J. (2009). A Survey on Metaheuristics for Stochastic Combinatorial Optimization. *Natural Comput*, *8*(2), 239–287. Doi: 10.1007/s11047-008-9098-4

[5]. Massan, S.-R., Wagan, A. I., & Shaikh, M. M. (2020). A New Metaheuristic Optimization Algorithm Inspired by Human Dynasties with an Application to the Wind Turbine Micrositing Problem. *Applied Soft Comput*ing, *90*, 106176. Doi: 10.1016/j.asoc.2020.106176

[6]. Wolpert, D. H., & Macready, W. G. (1997). No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67–82. Doi: 10.1109/4235.585893

[7]. Holland, J. H. (1975). Adaptation in Natural and Artificial Systems. *The University of Michigan Press: Ann Arbor*, *7*, 390-401.

[8]. Mahmood, M., & Al-Khateeb, B. (2019). The Blue Monkey: A New Nature Inspired Metaheuristic Optimization Algorithm. *Periodicals of Engineering and Natural Sciences (PEN)*, *7*(3), 1054. Doi: 10.21533/pen.v7i3.621

[9]. Ghasemi-Marzbali, A. (2020). A Novel Nature-Inspired Meta-Heuristic Algorithm for Optimization: Bear Smell Search Algorithm. *Soft computing*, *24*(17), 13003–13035. Doi: 10.1007/s00500-020-04721-1

[10]. Brammya, G., Praveena, S., Ninu Preetha, N. S., Ramya, R., Rajakumar, B. R., & Binu, D. (2019). Deer Hunting Optimization Algorithm: A New Nature-Inspired Meta-Heuristic Paradigm. *The Computer Journal*. Doi: 10.1093/comjnl/bxy133

[11]. Mohammadi-Balani, A., Dehghan Nayeri, M., Azar, A., & Taghizadeh-Yazdi, M. (2021). Golden Eagle Optimizer: A Nature-Inspired Metaheuristic Algorithm. *Computers & Industrial Engineering*, *152*, 107050. Doi: 10.1016/j.cie.2020.107050

[12]. Jain, M., Singh, V., & Rani, A. (2019). A Novel Nature-Inspired Algorithm for Optimization: Squirrel Search Algorithm. *Swarm and Evolutionary Computation*, *44*, 148–175. Doi: 10.1016/j.swevo.2018.02.013

[13]. Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A. A., Alqaness, M. A. A., & Gandomi, A. H. (2021). Aquila Optimizer: A Novel Meta-Heuristic Optimization Algorithm. *Computers & Industrial Engineering*, *157*, 107250. Doi: 10.1016/j.cie.2021.107250

[14]. Trojovský, P., & Dehghani, M. (2022). Pelican Optimization Algorithm: A Novel Nature-Inspired Algorithm for Engineering Applications. *Sensors*, *22*(3), 855. Doi: 10.3390/s22030855

[15]. Coufal, P., Hubálovský, Š., Hubálovská, M., & Balogh, Z. (2021). Snow Leopard Optimization Algorithm: A New Nature-Based Optimization Algorithm for Solving Optimization Problems. *Mathematics*, *9*(21), 2832. Doi: 10.3390/math9212832

[16]. Kirkpatrick, S., Gelatt Jr, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, *220*(4598), 671–680.

[17]. Eberhart, R., & Kennedy, J. A (1995). New Optimizer Using Particle Swarm Theory. *In MHS'95. Proceedings of the sixth international symposium on micro machine and human science*, 39–43.

[18]. Yang, X.S. (2009). Firefly Algorithms for Multimodal Optimization. *In Stochastic Algorithms: Foundations and Applications: 5th International Symposium*, SAGA, Sapporo, Japan, October 26-28. *Proceedings 5*, 169–178.

[19]. Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *IEEE computational intelligence magazine*, *1*(4), 28-39.

[20]. Karaboga, D., & Basturk, B. (2007). A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm. *Journal of Global Optimization*, *39*(3), 459–471. Doi: 10.1007/s10898-007-9149-x

[21]. Yang, X.S. (2010). A New Metaheuristic Bat-Inspired Algorithm. *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, 65–74. Doi: 10.1007/978-3-642-12538-6_6

[22]. Shan X., Jiang M., & Li J. (2006). The Routing Optimization Based on Improved Artificial Fish Swarm Algorithm. *In 6th World Congress on Intelligent Control and Automation*, *IEEE*, 3658–3662. Doi: 10.1109/WCICA.2006.1713052

[23]. Mucherino, A., Seref, O., Seref, O., Kundakcioglu, O. E., & Pardalos, P. (2007). Monkey Search: A Novel Metaheuristic Search for Global Optimization. *In AIP Conference Proceedings*, 162–173. Doi: 10.1063/1.2817338

[24]. Yang, X.S. (2012). Flower Pollination Algorithm for Global Optimization. *International Conference on Unconventional Computing and Natural Computation*, 240–249. Doi: 10.1007/978-3-642-32894-7_27

[25]. Pan, W.T. (2012). A New Fruit Fly Optimization Algorithm: Taking the Financial Distress Model as an Example. *Knowledge-Based Systems*, *26*, 69–74. Doi: 10.1016/j.knosys.2011.07.001

[26]. Gandomi, A. H., & Alavi, A. H. (2012). Krill Herd: A New Bio-Inspired Optimization Algorithm. *Communications in Nonlinear Science Numerical Simulation*, *17*(12), 4831–4845. Doi: 10.1016/j.cnsns.2012.05.010

[27]. Kaveh, A., & Farhoudi, N. (2013). A New Optimization Method: Dolphin Echolocation. *Advances in Engineering Software*, *59*, 53–70. Doi: 10.1016/j.advengsoft.2013.03.004

[28]. Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software*, *69*, 46–61. Doi: 10.1016/j.advengsoft.2013.12.007

[29]. Shareef, H., Ibrahim, A. A., & Mutlag, A. H. (2015). Lightning Search Algorithm. *Applied Soft Computing*, *36*, 315–333. Doi: 10.1016/j.asoc.2015.07.028

[30]. Uymaz, S. A., Tezel, G., & Yel, E. (2015). Artificial Algae Algorithm (AAA) for Nonlinear Global Optimization. *Applied Soft Computing*, *31*, 153–171. Doi: 10.1016/j.asoc.2015.03.003

[31]. Mani, M., Bozorg-Haddad, O., & Chu, X. (2018). Ant Lion Optimizer (ALO) *Algorithm*, 105–116. Doi: 10.1007/978-981-10-5221-7_11

[32]. Abedinia, O., Amjady, N., & Ghasemi, A. (2016). A New Metaheuristic Algorithm Based on Shark Smell Optimization. *Complexity*, *21*(5), 97–116. Doi: 10.1002/cplx.21634

[33]. Yong, W., Tao, W., Cheng-Zhi, Z., & Hua-Juan, H. (2016). A New Stochastic Optimization Approach — Dolphin Swarm Optimization Algorithm. *International Journal of Computational Intelligence and Applications*, *15*(2), 1650011. Doi: 10.1142/S1469026816500115

[34]. Mirjalili, S., & Lewis, A. (2016). The Whale Optimization Algorithm. *Advances in Engineering Software*, *95*, 51–67. Doi: 10.1016/j.advengsoft.2016.01.008

[35]. Abualigah, L., Shehab, M., Alshinwan, M., & Alabool, H. (2020). Salp swarm algorithm: a comprehensive survey. *Neural Computing and Applications*, *32*(15), 11195–11215. Doi: 10.1007/s00521-019-04629-4

[36]. Faramarzi, A., Heidarinejad, M., Mirjalili, S., & Gandomi, A. H. (2020). Marine Predators Algorithm: A Nature-Inspired Metaheuristic. *Expert Systens with Applications*, *152*, 113377. Doi: 10.1016/j.eswa.2020.113377

[37]. Li, M. D., Zhao, H., Weng, X. W., & Han, T. (2016). A Novel Nature-Inspired Algorithm for Optimization: Virus Colony Search. *Advances in Engineering Software*, *92*, 65–88. Doi: 10.1016/j.advengsoft.2015.11.004

[38]. Oyelade, O. N., Ezugwu, A. E.S., Mohamed, T. I. A., & Abualigah, L. (2022). Ebola Optimization Search Algorithm: A New Nature-Inspired Metaheuristic Optimization Algorithm. *IEEE Access*, *10*, 16150-16177. Doi: 10.1109/ACCESS.2022.3147821

[39]. Askarzadeh, A. (2016). A Novel Metaheuristic Method for Solving Constrained Engineering Optimization Problems: Crow Search Algorithm. *Computers & Structures*, *169*, 1–12. Doi: 10.1016/j.compstruc.2016.03.001

[40]. Yazdani, M., & Jolai, F. (2016). Lion Optimization Algorithm (LOA): A Nature-Inspired Metaheuristic Algorithm. *Journal of Computational Design Engineering*, *3*(1), 24–36. Doi: 10.1016/j.jcde.2015.06.003

[41]. Ghafori, S., & Gharehchopogh, F. S. (2022). Advances in Spotted Hyena Optimizer: A Comprehensive Survey. *Archives of Computational Methods in Engineering*, *29*(3), 1569–1590. Doi: 10.1007/s11831-021-09624-4

[42]. Abualigah, L., & Diabat, A. (2020). A comprehensive survey of the Grasshopper optimization algorithm: results, variants, and applications. *Neural Computing and Applications*, *32*(19), 15533–15556. Doi: 10.1007/s00521-020-04789-8

[43]. Qi, X., Zhu, Y., & Zhang, H. (2017). A New Meta-Heuristic Butterfly-Inspired Algorithm. *Journal of Computational Science*, *23*, 226–239. Doi: 10.1016/j.jocs.2017.06.003

[44]. Kaur, S., Awasthi, L. K., Sangal, A. L., & Dhiman, G. (2020). Tunicate Swarm Algorithm: A New Bio-Inspired Based Metaheuristic Paradigm for Global Optimization. *Engineering Applications of Artificial Intelligencel*, *90*, 103541. Doi: 10.1016/j.engappai.2020.103541

[45]. Seyyedabbasi, A., & Kiani, F. (2022). Sand Cat Swarm Optimization: A Nature-Inspired Algorithm to Solve Global Optimization Problems. *Engineering with Computers*. Doi: 10.1007/s00366-022-01604-x

[46]. Kifle, Z., Belay, G., & Bekele, A. (2013). Population Size, Group Composition and Behavioural Ecology of Geladas (Theropithecus Gelada) and Human-Gelada Conflict in Wonchit Valley, Ethiopia. *Pakistan Journal of Biological Sciences*, *16*(21), 1248–1259. Doi: 10.3923/pjbs.2013.1248.1259

[47]. Li, Z., & Hopfield, J. J. (1989). Modeling the Olfactory Bulb and Its Neural Oscillatory Processings. *Biological Cybernetics*, *61*(5), 379–392. Doi: 10.1007/BF00200803

[48]. Grossman, K. J., Mallik, A. K., Ross, J., Kay, L. M., & Issa, N. P. (2008). Glomerular Activation Patterns and the Perception of Odor Mixtures. *European Journal of Neuroscience*, *27*(10), 2676–2685. Doi: 10.1111/j.1460-9568.2008.06213.x

[49]. Jensi, R., & Jiji, G. W. (2016). An Enhanced Particle Swarm Optimization with Levy Flight for Global Optimization. *Applied Soft Computing*, *43*, 248–261. Doi: 10.1016/j.asoc.2016.02.018

[50]. Xie, J., Zhou, Y., & Chen, H. (2013). A Novel Bat Algorithm Based on Differential Operator and Lévy Flights Trajectory. *Computational Intelligence and Neuroscience*, *2013*, 1–13. Doi: 10.1155/2013/453812

[51]. Fister Jr, I., Mlakar, U., Brest, J., & Fister, I. (2016). A new population-based nature-inspired algorithm every month: is the current era coming to the end. In *Proceedings of the 3rd student computer science research conference*, 33-37. University of Primorska Press.