MDPI

Article

EEGO: an extended version of Eel and grouper optimizer for global optimization problems

Glykeria Kyrou[1], Vasileios Charilogis[2] and Ioannis G. Tsoulos[3,*]

[1] Department of Informatics and Telecommunications, University of Ioannina, 47150 Kostaki Artas, Greece; g.kyrou@uoi.gr
[2] Department of Informatics and Telecommunications, University of Ioannina, 47150 Kostaki Artas, Greece; v.charilog@uoi.gr
[3] Department of Informatics and Telecommunications, University of Ioannina, 47150 Kostaki Artas, Greece;itsoulos@uoi.gr
* Correspondence: itsoulos@uoi.gr

Abstract: The problems of finding a global minimum of a function are increasingly applied to real-world problems. As a result, a variety of computational techniques have been developed to better locate the global minimum. A decisive role is played by evolutionary techniques, which simulate natural processes and aim to find the global minimum of multidimensional functions. A recently introduced evolutionary technique is the optimal Eel and Grouper (EGO) algorithm, which is inspired by the symbiotic interaction and foraging strategy of eels and groupers in marine ecosystems. The EGO algorithm is characterized for its reliability in locating the global minimum. In this paper, modifications are proposed that aim to improve the reliability and speed of the above technique, such as the application of a termination technique based on stochastic observations and an innovative sampling method. The proposed method was tested on several problems from the relevant literature and a comparative study was made with other global optimization techniques with promising results.

Keywords: Global optimization; Meta-heuristic; Stochastic techniques; Swarm based methods

1. Introduction

The basic goal of global optimization is to find the global minimum by searching for the appropriate scope of each problem. Primarily, a global optimization method aims to discover the global minimum of a continuous function, and it is defined as

$$x^* = \arg\min_{x \in S} f(x) \tag{1}$$

with S:

$$S = [a_1, b_1] \times [a_2, b_2] \times \ldots [a_n, b_n]$$

Global optimization is an integral part of many areas of our daily lives. One of these fields is computer science[9,50,51]. In addition to the computer field, global optimization finds application in sciences such as mathematics[8,52–54], physics[10–12], chemistry[13–15] and medicine[16,17,56]. Optimization methods are divided into two categories, deterministic[61–63] and stochastic[64–66]. The first category, i.e. that of deterministic methods, has as its main objective the identification of the overall optimal solution and is mainly used in simple problems. On the contrary, stochastic methods are mainly used in complex problems.

Swarm Intelligence Algorithms[48,49,55,69,70] as a source of inspiration and collective behavior of insects and other animals. Intelligence Algorithms mimic systems in which agents interact locally and cooperate worldwide to solve optimization problems. Swarm intelligence algorithms are very important tools for dealing with complex optimization problems in a wide range of applications[27]. Characteristic examples of such algorithms are

Citation: Kyrou, G.; Charilogis, V.; Tsoulos, I.G. EEGO: an extended version of Eel and grouper optimizer for global optimization problems. *Journal Not Specified* **2024**, *1*, 0. https://doi.org/

Received:
Revised:
Accepted:
Published:

the WOA algorithm 28,31,34,57–59, SCA algorithm [29,33,36,60,68]and SSA algorithm[30, 32,35,58,60].

The EGO algorithm is a reliable technique for real-world optimization problems. The EGO algorithm is inspired by the symbiotic interaction and foraging strategy of eels and groupers in marine ecosystems. In nature, there are complex interactions between biological species. Relationships of this type are classified into five categories[1,2]:1. Naturalism: Where two species can live in an ecosystem without affecting each other. 2. In predation, where one creature dies to feed another. 3. Parasitism: where one species causes harm to another without always killing it. 4. In competitive mode, the same or different organizations compete for resources. 5. Mutualism: [4–6]when two organisms have a beneficial INTERACTION. Bshary et al. [3]consider that target ingestion, something observed in eels and groupers, is a necessary condition for interspecific cooperative hunting to occur. Intraspecific predation could increase the hunting efficiency of predators by mammals. According to Ali Mohammadzadeh and Seyedali Mirjalili the EGO optimization algorithm [7]generates a set of random answers, then stores the best answers found so far, allocates them to the target point, and changes the answers with them. As the number of iterations increases, the limits of the sine function are changed to enhance the phase of finding the best solution. This method stops the process when the iteration exceeds the maximum number. Because the EGO optimization algorithm generates and boosts a collection of random responses, it has the advantage of increased local optimum discovery and avoidance compared to individual methods.

This paper introduces some modifications to the EGO algorithm in order to improve its efficiency. The proposed amendments are presented below:

- The addition of a sampling technique based on the K-means method[37,42,44–47]. The sampled points will help to find the global minimum of the function in the most efficient way. Additionally, by using this method, points that are in proximity can be rejected.
- The use of a termination technique based on stochastic observations. At each iteration of the algorithm, the smallest value is recorded. Once it remains stable for a predetermined number of iterations, the method terminates. The present termination method helps with the fastest termination without unnecessarily wasting computing time.
- Using mod1 and mod2 and mod3 parameters which take values of either 1, which means it is Disable or 2, which means it is Enable. mod1 specifies how randomness will be used when updating solutions. When mod1 is set to 1 p, it takes values from 0 to 1, while when mod1 is set to 2 p it takes values from -1.0 to 1.0. The mod2 parameter affects how the variables f1 and f2 are defined, which in turn affects the calculation of the new positions. The mod3 parameter determines how the algorithm handles thresholds when agent positions exceed predefined minimum and maximum thresholds. When mod3 is set to 1, it ensures that all positions remain within valid limits. When mod3 is set to 2 these positions are not taken into account and the fish wait for better positions to attack.

The rest of this paper is divided into the following sections: in section 2, the proposed method is fully described.

2. The proposed method

2.1. The main steps of the algorithm

The main steps of the used global optimization method are the following:

1. **Initialization step**.
   (a) Define as N_c the number of elements in the search Agents
   (b) Define as N_g, the maximum number of allowed iterations.
   (c) Initialize randomly the members of the search Agents in set S.
2. **Calculation step**.
   (a) While $[(t \prec Ng)]$

(b) Update Variables a and starrvation_rate where starrvation_rate is a number in [0,100]. According to types:

- $a = 2 - 2 * (t \ / \ Ng)$
- $(starvatiorate = 100 * (t / \ Ng))$

(c) Compute the fitness of each search Agents.

(d) Sort all solutions in the current population fro the best to worst according to the function value.

For $j = 1, \ldots, N_c$ **do**

- Update variables r1, r2, r3, r4, C1, C2 and p : where r_1, r_2 are random numbers in , $[0, 1]$, C1 is a rondom number in [-a,a], C2 is a random number in [0,2]. According to types:
  - $r3 = (a - 2) * r1 + 2$
  - $r4 = 100 * r2$
  - $C1 = 2 * a * r1 - a$
  - $C2 = 2 * r1$
  - $b = a * r2$
- Update the position of each search Agent:
- $if(r4 <= starvation_rate)$
- if mod1 has the value 1
  - p is a random number in $[0.0, 1.0]$
- if mod1 has the value 2
  - p is a random number in $[-1.0, 1.0]$
- if mod2 has the value 1
  - $f1 = 0.8$
  - $f2 = 0.2$
- if mod2 has the value 2
  - f1 is a random number in $[0.0, 2.0]$
  - f1 is a random number in $[-2.0, 0.0]$

If $(p \prec 0.5)$

- Change the location of current search by :$[(f1 * X1 \dot{+} f2 * X2) \ /2]$

else if $(p \geq 0.5)$

- Change the location of current search by :$(f2 * X1 \dot{+} f1 * X2 \ /2)$

End if

End For

(e) Ensure No Agents leave the search area

(f) Evaluate the performance of each search Agents

(g) Update XPrey

- if there is a better solution

(h) Set $t = t + 1$

(i) End while

(j) Return the best solution XPrey

3. Termination check step

(a) Set $t = t + 1$

(b) Calculate to stopping rule in the work of Charilogis [43]

(c) If the termination criteria are not met then go to Calculation step, **else** terminate.

2.2. The proposed sampling procedure

The sampling technique applied in this work initially generates samples from the objective function. Then, through the K-means method, only the recognized centers are selected as final samples. This technique, which is an achievement of James MacQueen [41], is one of the most well-known clustering algorithms in the broad research community, both in data analysis and in machine learning [42, 43] and pattern recognition[46]. The algorithm mainly aims to estimate the centers of possible groups from a set of samples. Next, the basic steps of the algorithm are presented:

1. **Define** as k the number of clusters.
2. Randomly select N_m initial points $x_i,\ i = 1, \ldots, N_m$ from the objective function.
3. Randomly assign each point $x_i,\ i = 1, ..., N_m$ in a cluster $S_j,\ j = 1, \ldots, k$.
4. **For** every cluster $j = 1..k$ **do**
   - **Set** as M_j the number of points in S_j
   - **Calculate** the center of the cluster c_j as

$$c_j = \frac{1}{M_j} \sum_{x_i \in S_j} x_i$$

5. **End For**
6. **Repeat the following steps:**
   - Set $S_j = \{\},\ j = 1..k$
   - **For** each point $x_i,\ i = 1, ..., N_m$ **do**
     - **Set** $j^* = \text{argmin}_{m=1}^{k}\{D(x_i, c_m)\}$. The function $D(x, y)$ is the Euclidean distance of points (x, y).
     - **Set** $S_{j^*} = S_{j^*} \cup \{x_i\}$.
   - **End For**
   - **For** each center $c_j,\ j = 1..k$ **do**
     - **Update** the center c_j as

$$c_j = \frac{1}{M_j} \sum_{x_i \in S_j} x_i$$

   - **End For**
7. **If** there is no significant change in centers c_j **terminate** the algorithm and return the k centers as the final set of samples.

3. Results

This section will begin with a detailed description of the functions that will be used in the experiments, followed by an analysis of the experiments performed and comparisons with other global optimization techniques.

3.1. Test functions

The functions used in the experiments have been proposed in a series of relative works[18,19] and they cover various scientific fields, such as medicine, physics, engineering, etc. Also, these objective functions have been used by many researchers in a variety of publications.[20–24] The definitions of these functions are given below:

- **Bf1** (Bohachevsky 1) function:

$$f(x) = x_1^2 + 2x_2^2 - \frac{3}{10}\cos(3\pi x_1) - \frac{4}{10}\cos(4\pi x_2) + \frac{7}{10}$$

- **Bf2** (Bohachevsky 2) function:

$$f(x) = x_1^2 + 2x_2^2 - \frac{3}{10}\cos(3\pi x_1)\cos(4\pi x_2) + \frac{3}{10}$$

- **Bf3** (Bohachevsky 3) function:

$$f(x) = x_1^2 + 2x_2^2 - \frac{3}{10}\cos(3\pi x_1 + 4\pi x_2) + \frac{3}{10}$$

- **Branin** function:

$$f(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10$$

with $-5 \leq x_1 \leq 10,\ 0 \leq x_2 \leq 15$.

- **Camel** function:

$$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4, \quad x \in [-5,5]^2$$

- **Easom** function:

$$f(x) = -\cos(x_1)\cos(x_2)\exp\left((x_2-\pi)^2 - (x_1-\pi)^2\right)$$

with $x \in [-100,100]^2$.

- **Exponential** function, defined as:

$$f(x) = -\exp\left(-0.5\sum_{i=1}^{n} x_i^2\right), \quad -1 \leq x_i \leq 1$$

In the conducted experiments the values $n = 4, 8, 16, 32$ were used.

- **Griewank2** function:

$$f(x) = 1 + \frac{1}{200}\sum_{i=1}^{2} x_i^2 - \prod_{i=1}^{2}\frac{\cos(x_i)}{\sqrt{(i)}}, \quad x \in [-100,100]^2$$

- **Griewank10** function. The function is given by the equation

$$f(x) = \sum_{i=1}^{n}\frac{x_i^2}{4000} - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

with $n = 10$.

- **Gkls** function[25]. $f(x) = \text{Gkls}(x, n, w)$, is a constructed function with w local minima presented in [25], with $x \in [-1,1]^n$. For the conducted experiments the values $n = 2, 3$ and $w = 50$ were utilized.

- **Goldstein and Price function**

$$\begin{aligned} f(x) \quad = \quad & \Big[1 + (x_1 + x_2 + 1)^2 \\ & \left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2\right)\Big] \times \\ & [30 + (2x_1 - 3x_2)^2 \\ & \left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2\right)] \end{aligned}$$

With $x \in [-2,2]^2$.

- **Hansen** function: $f(x) = \sum_{i=1}^{5} i\cos[(i-1)x_1 + i] \sum_{j=1}^{5} j\cos[(j+1)x_2 + j]$, $x \in [-10,10]^2$.
- **Hartman 3** function:

$$f(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right)$$

with $x \in [0,1]^3$ and $a = \begin{pmatrix} 3 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3 & 10 & 30 \\ 0.1 & 10 & 35 \end{pmatrix}$, $c = \begin{pmatrix} 1 \\ 1.2 \\ 3 \\ 3.2 \end{pmatrix}$ and

$$p = \begin{pmatrix} 0.3689 & 0.117 & 0.2673 \\ 0.4699 & 0.4387 & 0.747 \\ 0.1091 & 0.8732 & 0.5547 \\ 0.03815 & 0.5743 & 0.8828 \end{pmatrix}$$

- **Hartman 6** function:

$$f(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2\right)$$

with $x \in [0,1]^6$ and $a = \begin{pmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{pmatrix}$, $c = \begin{pmatrix} 1 \\ 1.2 \\ 3 \\ 3.2 \end{pmatrix}$ and

$$p = \begin{pmatrix} 0.1312 & 0.1696 & 0.5569 & 0.0124 & 0.8283 & 0.5886 \\ 0.2329 & 0.4135 & 0.8307 & 0.3736 & 0.1004 & 0.9991 \\ 0.2348 & 0.1451 & 0.3522 & 0.2883 & 0.3047 & 0.6650 \\ 0.4047 & 0.8828 & 0.8732 & 0.5743 & 0.1091 & 0.0381 \end{pmatrix}$$

- **Potential** function, this function stands for the energy of a molecular conformation of N atoms, that interacts using via the Lennard-Jones potential[26]. The function is defined as:

$$V_{LJ}(r) = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$$

For the conducted experiments the values $N = 3,\ 5$ were used.

- **Rastrigin** function.

$$f(x) = x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2), \quad x \in [-1,1]^2$$

- ***Rosenbrock*** *function*.

$$f(x) = \sum_{i=1}^{n-1}\left(100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right), \quad -30 \leq x_i \leq 30.$$

The values $n = 4,\ 8,\ 16$ were used in the conducted experiments.

- **Shekel 5** function.

$$f(x) = -\sum_{i=1}^{5} \frac{1}{(x - a_i)(x - a_i)^T + c_i}$$

with $x \in [0,10]^4$ and $a = \begin{pmatrix} 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 \\ 8 & 8 & 8 & 8 \\ 6 & 6 & 6 & 6 \\ 3 & 7 & 3 & 7 \end{pmatrix}$, $c = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.2 \\ 0.4 \\ 0.4 \end{pmatrix}$

- **Shekel 7** function.

$$f(x) = -\sum_{i=1}^{7} \frac{1}{(x-a_i)(x-a_i)^T + c_i}$$

with $x \in [0,10]^4$ and $a = \begin{pmatrix} 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 \\ 8 & 8 & 8 & 8 \\ 6 & 6 & 6 & 6 \\ 3 & 7 & 3 & 7 \\ 2 & 9 & 2 & 9 \\ 5 & 3 & 5 & 3 \end{pmatrix}$, $c = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.2 \\ 0.4 \\ 0.4 \\ 0.6 \\ 0.3 \end{pmatrix}$.

- **Shekel 10** function.

$$f(x) = -\sum_{i=1}^{10} \frac{1}{(x-a_i)(x-a_i)^T + c_i}$$

with $x \in [0,10]^4$ and $a = \begin{pmatrix} 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 \\ 8 & 8 & 8 & 8 \\ 6 & 6 & 6 & 6 \\ 3 & 7 & 3 & 7 \\ 2 & 9 & 2 & 9 \\ 5 & 5 & 3 & 3 \\ 8 & 1 & 8 & 1 \\ 6 & 2 & 6 & 2 \\ 7 & 3.6 & 7 & 3.6 \end{pmatrix}$, $c = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.2 \\ 0.4 \\ 0.4 \\ 0.6 \\ 0.3 \\ 0.7 \\ 0.5 \\ 0.6 \end{pmatrix}$.

- **Sinusoidal** function defined as:

$$f(x) = -\left(2.5\prod_{i=1}^{n}\sin(x_i - z) + \prod_{i=1}^{n}\sin(5(x_i - z))\right), \quad 0 \leq x_i \leq \pi.$$

The values of $n = 4, 8, 16$ were used in the conducted experiments.

- **Test2N** function:

$$f(x) = \frac{1}{2}\sum_{i=1}^{n} x_i^4 - 16x_i^2 + 5x_i, \quad x_i \in [-5, 5].$$

For the conducted experiments the values $n = 4, 5, 6, 7$ were used.

- **Test30N** function:

$$f(x) = \frac{1}{10}\sin^2(3\pi x_1)\sum_{i=2}^{n-1}\Big((x_i-1)^2\Big(1+\sin^2(3\pi x_{i+1})\Big)\Big) + (x_n-1)^2\Big(1+\sin^2(2\pi x_n)\Big)$$

The values $n = 3, 4$ were used in the conducted experiments.

3.2. Experimental results

The following applies to table 2:

- The column FUNCTION denotes the name of the objective problem.

Table 1. Experimental settings. The numbers in cells denote the values used in the experiments for all parameters.

Nc	Number of chromosomes/particles	200
Ng	Maximum number of allowed iterations	200
Nm	Number of initial samples for K-means	10 × Nc
Nk	Number of iterations for stopping rule	5
ps	Selection rate for the genetic algorithm	0.1
pm	Mutation rate for the genetic algorithm	0.05

- The column GENETIC denotes the application of a genetic algorith to the objective problem. The genetic algorithm has N_c chromosomes and the maximum number of allowed generations was set to N_g.
- The column PSO stands for the application of Particle Swarm Optimizer to every objective problem. The number of particles was set to N_c and the maximum number of allowed iterations was set to N_g.
- The column EEGO represents the application of the proposed method using the values for the parameters shown in Table
- The row SUM represents the sum of function calls for all test functions.

Table 2. Experimental results using different optimization methods. Numbers in cells represent sum function calls.

FUNCTION	GENETIC	iPSO	DE	GAO	EOFA
BF1	4007	4142	8268	4139	3228
BF2	3794	3752	7913	3775	2815
BRANIN	2376	2548	4101	2353	1684
CAMEL	2869	2933	5609	2856	2262
EASOM	1958	1982	2978	1972	1334
EXP4	2946	3404	5166	2918	2166
EXP8	3120	3585	5895	2989	2802
EXP16	3250	3735	6498	3110	3279
EXP32	3561	3902	7606	3319	3430
GKLS250	2280	2411	3834	2435	1603
GKLS350	2613	2234	3919	2773	1298
GOLDSTEIN	3687	3865	6781	3859	2784
GRIEWANK2	4501	3076	7429	4174	2589
GRIEWANK10	6410	8006	18490	7577	7435
HANSEN	3210	2856	4185	3506	2484
HARTMAN3	2752	3140	5190	2951	1793
HARTMAN6	3219	3710	5968	3635	2478
POTENTIAL3	4352	4865	6118	3862	4081
POTENTIAL5	7705	9183	9119	6883	8886
RASTRIGIN	4107	3477	6216	3945	2304
ROSENBROCK4	3679	6372	8452	5115	4019
ROSENBROCK8	5270	8284	11530	6857	6801
ROSENBROCK16	8509	11872	17432	9862	11996
SHEKEL5	3325	4259	6662	3645	2495
SHEKEL7	3360	4241	6967	3674	2432
SHEKEL10	3488	4237	6757	3665	2516
TEST2N4	3331	3437	6396	3423	2277
TEST2N5	4000	3683	6271	3889	2734
TEST2N6	4312	3781	5410	4267	2905
TEST2N7	4775	4060	7074	4403	3559
SINU4	2991	3504	5953	3265	2005
SINU8	3442	4213	6973	3844	3158
SINU16	4320	5019	6979	4813	5891
TEST30N3	3211	4610	6168	3864	2362
TEST30N4	3679	4629	7006	4772	2978
SUM	134409	153006	247313	142389	118863

One more experiment which was performed with the ultimate goal of measuring the importance of K-means sampling in the proposed method. 2 different sampling methods were used in this experiment:

- The UNIFORM column is about incorporating uniform sampling into the genetic algorithm. Where N_{c} are randomly selected chromosomes using uniform sampling in the genetic algorithm.
- The KMEANS column, which refers to the implementation of k-means sampling as proposed in the genetic algorithm.

Table 3. Experiments using different sampling techniques for the proposed method.

FUNCTION	UNIFORM	KMEANS
BF1	4513	3228
BF2	3959	2815
BRANIN	2282	1684
CAMEL	3156	2262
EASOM	1756	1334
EXP4	3438	2166
EXP8	3432	2802
EXP16	3369	3279
EXP32	3216	3430
GKLS250	2268	1603
GKLS350	2151	1298
GOLDSTEIN	3855	2784
GRIEWANK2	4310	2589
GRIEWANK10	8640	7435
HANSEN	3329	2484
HARTMAN3	2849	1793
HARTMAN6	3456	2478
POTENTIAL3	4554	4081
POTENTIAL5	8356	8886
RASTRIGIN	3310	2304
ROSENBROCK4	6566	4019
ROSENBROCK8	8379	6801
ROSENBROCK16	11921	11996
SHEKEL5	3946	2495
SHEKEL7	3990	2432
SHEKEL10	3836	2516
TEST2N4	3345	2277
TEST2N5	3937	2734
TEST2N6	4008	2905
TEST2N7	4545	3559
SINU4	3128	2005
SINU8	4126	3158
SINU16	6774	5891
TEST30N3	3704	2362
TEST30N4	4262	2978
SUM	152666	118863

4. Conclusions

In this particular article, some modifications were proposed to the EEGO optimization method, which had the main purpose of improving the efficiency as well as the speed of the global optimization algorithm. The first modification was to periodically apply a sampling technique based on the K-Means method[47] Using the sampling method we proposed helped to find the global minimum with the greatest accuracy and in the least possible time. The second proposed modification concerns the termination rule, which helps to avoid unnecessary computational time being wasted in iterations.

Because the experimental results are very promising, efforts can be made to develop the technique in various fields. A future extension of the application may be the use of parallel computing techniques to speed up the optimization process, such as, for example, the integration of MPI[38] or the OpenMP library[39].

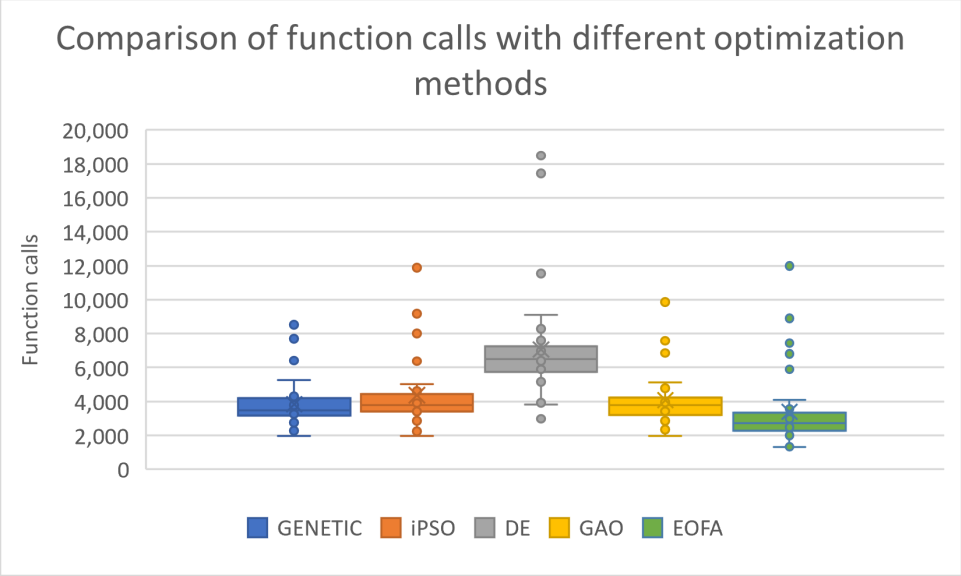

Figure 1. xxxxx

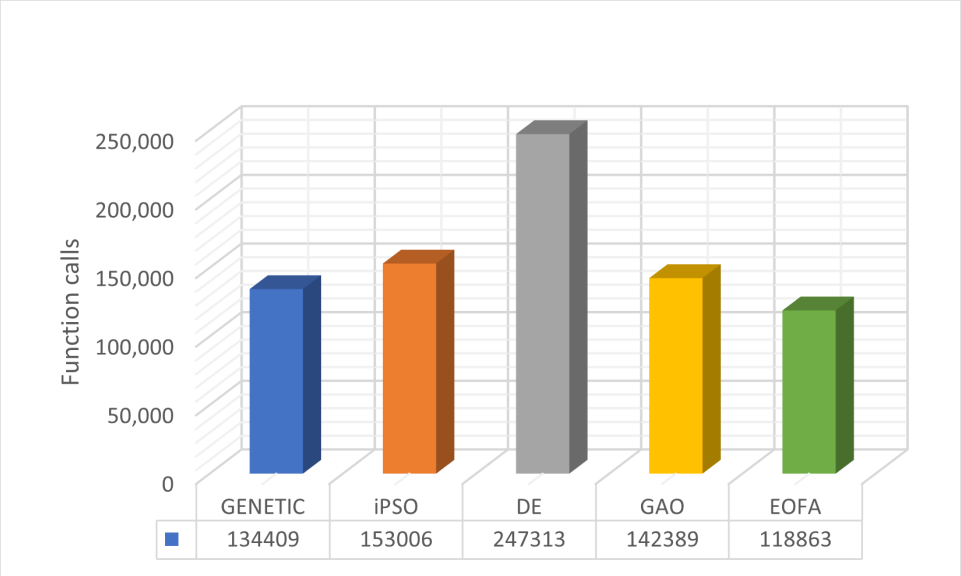

Figure 2. xxxxx

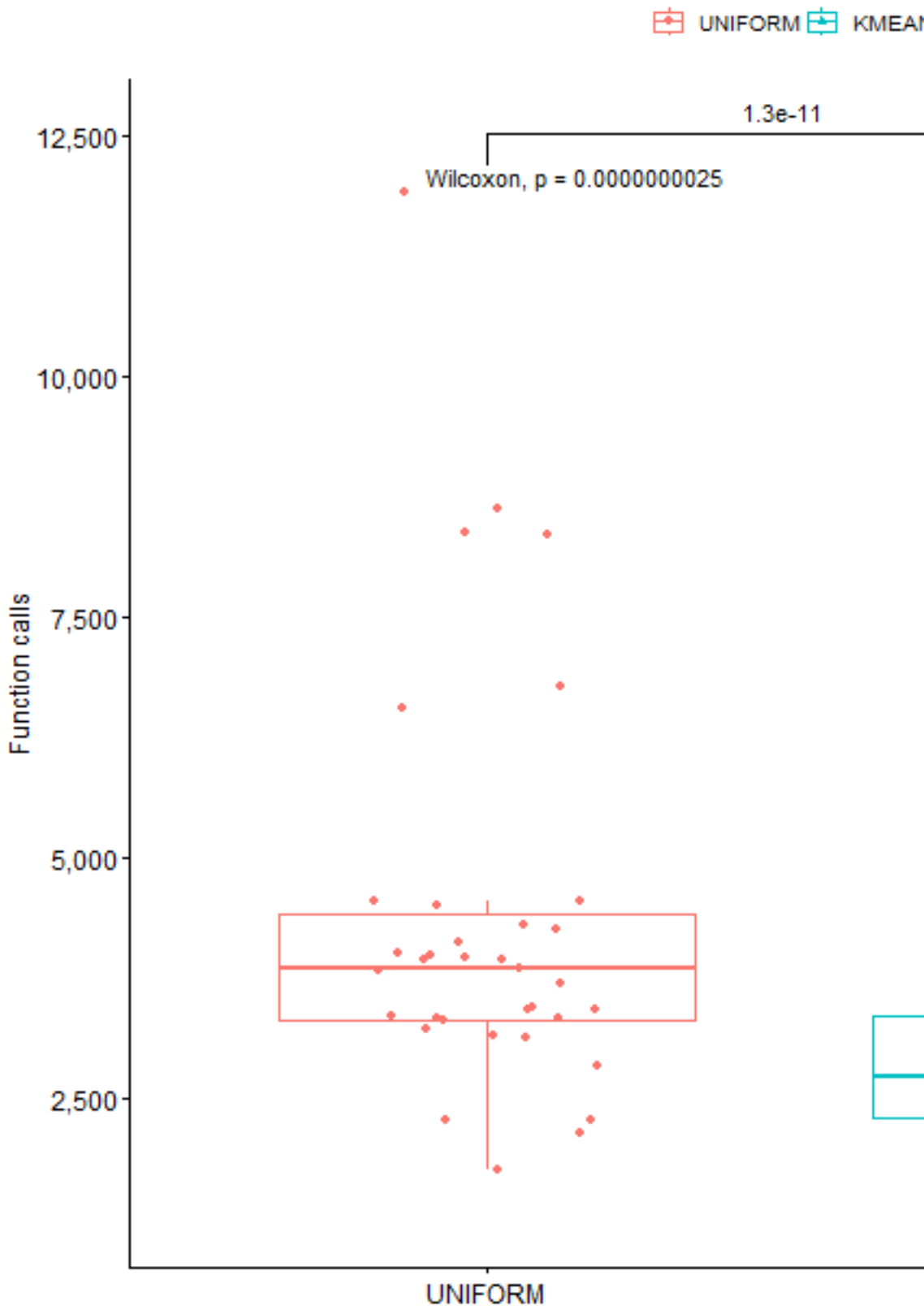

Figure 3. xxxxxx

Author Contributions: For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used "Conceptualization, X.X. and Y.Y.; methodology, X.X.; software, X.X.; validation, X.X., Y.Y. and Z.Z.; formal analysis, X.X.; investigation, X.X.; resources, X.X.; data curation, X.X.; writing—original draft preparation, X.X.; writing—review and editing, X.X.; visualization, X.X.; supervision, X.X.; project administration, X.X.; funding acquisition, Y.Y. All authors have read and agreed to the published version of the manuscript.", please turn to the CRediT taxonomyfor the term explanation. Authorship must be limited to those who have contributed substantially to the work reported.

Funding: Please add: "This research received no external funding" or "This research was funded by NAME OF FUNDER grant number XXX." and and "The APC was funded by XXX". Check carefully that the details given are accurate and use the standard spelling of funding agency names at https://search.crossref.org/funding, any errors may affect your future funding.

Institutional Review Board Statement: In this section, you should add the Institutional Review Board Statement and approval number, if relevant to your study. You might choose to exclude this statement if the study did not require ethical approval. Please note that the Editorial Office might ask you for further information. Please add "The study was conducted in accordance with the Declaration of Helsinki, and approved by the Institutional Review Board (or Ethics Committee) of NAME OF INSTITUTE (protocol code XXX and date of approval)." for studies involving humans. OR "The animal study protocol was approved by the Institutional Review Board (or Ethics Committee) of NAME OF INSTITUTE (protocol code XXX and date of approval)." for studies involving animals. OR "Ethical review and approval were waived for this study due to REASON (please provide a detailed justification)." OR "Not applicable" for studies not involving humans or animals.

Informed Consent Statement: Any research article describing a study involving humans should contain this statement. Please add "Informed consent was obtained from all subjects involved in the study." OR "Patient consent was waived due to REASON (please provide a detailed justification)." OR "Not applicable" for studies not involving humans. You might also choose to exclude this statement if the study did not involve humans.

Written informed consent for publication must be obtained from participating patients who can be identified (including by the patients themselves). Please state "Written informed consent has been obtained from the patient(s) to publish this paper" if applicable.

Data Availability Statement: We encourage all authors of articles published in MDPI journals to share their research data. In this section, please provide details regarding where data supporting reported results can be found, including links to publicly archived datasets analyzed or generated during the study. Where no new data were created, or where data is unavailable due to privacy or ethical re-strictions, a statement is still required. Suggested Data Availability Statements are available in section "MDPI Research Data Policies" at https://www.mdpi.com/ethics.

Acknowledgments: In this section you can acknowledge any support given which is not covered by the author contribution or funding sections. This may include administrative and technical support, or donations in kind (e.g., materials used for experiments).

Conflicts of Interest: Declare conflicts of interest or state "The authors declare no conflicts of interest." Authors must identify and declare any personal circumstances or interest that may be perceived as inappropriately influencing the representation or interpretation of reported research results. Any role of the funders in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; or in the decision to publish the results must be declared in this section. If there is no role, please state "The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results".

References

1. Usman, M.J.: A survey of symbiotic organisms search algorithms and applications. Neural Comput. Appl. 32(2), 547–566 (2020)
2. . Ezugwu, A.E., Adeleke, O.J., Akinyelu, A.A., Viriri, S.: A conceptual comparison of several metaheuristic algorithms on continuous optimisation problems. Neural Comput. Appl. 32(10), 6207–6251 (2020)
3. Bshary, R., Hohner, A., Ait-el-Djoudi, K., Fricke, H.: Interspecific communicative and coordinated hunting between groupers and giant moray eels in the Red Sea. PLoS Biol. 4(12), e431 (2006)

4. Wang, Y., & DeAngelis, D. L. (2012). A mutualism-parasitism system modeling host and parasite withmutualism at low density. Mathematical Biosciences & Engineering, 9(2), 431-444.
5. Addicott, J. F. (1985). Competition in mutualistic systems. The biology of mutualism: ecology and evolution. Croom Helm, London, UK, 217-247.
6. Aubier, T. G., Joron, M., & Sherratt, T. N. (2017). Mimicry among unequally defended prey should be mutualistic when predators sample optimally. The American Naturalist, 189(3), 267-282.
7. Ali Mohammadzadeh, Seyedali Mirjalili. (2024) Eel and grouper optimizer: a nature-inspired optimization algorithm. Springer Science+Business Media, LLC, part of Springer Nature 2024
8. Intriligator, M. D. (2002). Mathematical optimization and economic theory. Society for Industrial and Applied Mathematics.
9. A. Törn, M.M. Ali, S. Viitanen, Stochastic global optimization: Problem classes and solution techniques. Journal of Global Optimization **14**, pp. 437-447, 1999.
10. L. Yang, D. Robin, F. Sannibale, C. Steier, W. Wan, Global optimization of an accelerator lattice using multiobjective genetic algorithms, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **609**, pp. 50-57, 2009.
11. E. Iuliano, Global optimization of benchmark aerodynamic cases using physics-based surrogate models, Aerospace Science and Technology **67**, pp.273-286, 2017.
12. Q. Duan, S. Sorooshian, V. Gupta, Effective and efficient global optimization for conceptual rainfall-runoff models, Water Resources Research **28**, pp. 1015-1031 , 1992.
13. S. Heiles, R. L. Johnston, Global optimization of clusters using electronic structure methods, Int. J. Quantum Chem. **113**, pp. 2091–2109, 2013.
14. W.H. Shin, J.K. Kim, D.S. Kim, C. Seok, GalaxyDock2: Protein–ligand docking using beta-complex and global optimization, J. Comput. Chem. **34**, pp. 2647– 2656, 2013.
15. A. Liwo, J. Lee, D.R. Ripoll, J. Pillardy, H. A. Scheraga, Protein structure prediction by global optimization of a potential energy function, Biophysics **96**, pp. 5482-5485, 1999.
16. Eva K. Lee, Large-Scale Optimization-Based Classification Models in Medicine and Biology, Annals of Biomedical Engineering **35**, pp 1095-1109, 2007.
17. Y. Cherruault, Global optimization in biology and medicine, Mathematical and Computer Modelling **20**, pp. 119-132, 1994.
18. M. Montaz Ali, Charoenchai Khompatraporn, Zelda B. Zabinsky, A Numerical Evaluation of Several Stochastic Algorithms on Selected Continuous Global Optimization Test Problems, Journal of Global Optimization **31**, pp 635-672, 2005.
19. C.A. Floudas, P.M. Pardalos, C. Adjiman, W. Esposoto, Z. Gümüs, S. Harding, J. Klepeis, C. Meyer, C. Schweiger, Handbook of Test Problems in Local and Global Optimization, Kluwer Academic Publishers, Dordrecht, 1999.
20. M.M. Ali and P. Kaelo, Improved particle swarm algorithms for global optimization, Applied Mathematics and Computation **196**, pp. 578-593, 2008.
21. H. Koyuncu, R. Ceylan, A PSO based approach: Scout particle swarm algorithm for continuous global optimization problems, Journal of Computational Design and Engineering **6**, pp. 129–142, 2019.
22. Patrick Siarry, Gérard Berthiau, François Durdin, Jacques Haussy, ACM Transactions on Mathematical Software **23**, pp 209–228, 1997.
23. I.G. Tsoulos, I.E. Lagaris, GenMin: An enhanced genetic algorithm for global optimization, Computer Physics Communications **178,** pp. 843-851, 2008.
24. A. LaTorre, D. Molina, E. Osaba, J. Poyatos, J. Del Ser, F. Herrera, A prescription of methodological guidelines for comparing bio-inspired optimization algorithms, Swarm and Evolutionary Computation **67**, 100973, 2021.
25. M. Gaviano, D.E. Ksasov, D. Lera, Y.D. Sergeyev, Software for generation of classes of test functions with known local and global minima for global optimization, ACM Trans. Math. Softw. **29**, pp. 469-480, 2003.
26. J.E. Lennard-Jones, On the Determination of Molecular Fields, Proc. R. Soc. Lond. A **106**, pp. 463–477, 1924.
27. Tang, J., Liu, G., Pan, Q.: A review on representative swarm intelligence algorithms for solving optimization problems:applications and trends. IEEE/CAA J. Autom. Sin. 8(10),1627–1643 (2021)
28. Mirjalili, S., Lewis, A.: The whale optimization algorithm. Adv. Eng. Softw. 95, 51–67 (2016)
29. Mirjalili, S.: SCA: a sine cosine algorithm for solving optimization problems. Knowl.-Based Syst. 96, 120–133 (2016)
30. Mirjalili, S., Gandomi, A.H., Mirjalili, S.Z., Saremi, S., Faris, H., Mirjalili, S.M.: Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. Adv. Eng. Softw. 114, 163–191 (2017)
31. Nasiri, J., & Khiyabani, F. M. (2018). A whale optimization algorithm (WOA) approach for clustering. Cogent Mathematics & Statistics, 5(1), 1483565.
32. Bairathi, D., & Gopalani, D. (2019). Salp swarm algorithm (SSA) for training feed-forward neural networks. In Soft Computing for Problem Solving: SocProS 2017, Volume 1 (pp. 521-534). Springer Singapore.
33. Hao, Y., Song, L., Cui, L., & Wang, H. (2019). A three-dimensional geometric features-based SCA algorithm for compound faults diagnosis. Measurement, 134, 480-491.
34. Gharehchopogh, F. S., & Gholizadeh, H. (2019). A comprehensive survey: Whale Optimization Algorithm and its applications. Swarm and Evolutionary Computation, 48, 1-24.

35. Abualigah, L., Shehab, M., Alshinwan, M., & Alabool, H. (2020). Salp swarm algorithm: a comprehensive survey. Neural Computing and Applications, 32(15), 11195-11215.
36. Sahu, P. C., Prusty, R. C., & Panda, S. (2022). Optimal design of a robust FO-Multistage controller for the frequency awareness of an islanded AC microgrid under i-SCA algorithm. International Journal of Ambient Energy, 43(1), 2681-2693.
37. Powell, M.J.D. A Tolerant Algorithm for Linearly Constrained Optimization Calculations. Math. Program. 1989, 45, 547–566.
38. Gropp, W.; Lusk, E.; Doss, N.; Skjellum, A. A high-performance, portable implementation of the MPI message passing interface standard. Parallel Comput. 1996, 22, 789–828.
39. Chandra, R. Parallel Programming in OpenMP; Morgan Kaufmann: Cambridge, MA, USA, 2001.
40. J.B. MacQueen, Some Methods for classification and Analysis of Multivariate Observations. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. Vol. 1. University of California Press. pp. 281–297. MR 0214227. Zbl 0214.46201, 1967.
41. Y. Li, H. Wu, A clustering method based on K-means algorithm, Physics Procedia **25**, pp. 1104-1109, 2012.
42. P. Arora, S. Varshney, Analysis of k-means and k-medoids algorithm for big data, Procedia Computer Science **78**, pp. 507-512, 2016.
43. Charilogis, V.; Tsoulos, I.G. Toward an Ideal Particle Swarm Optimizer for Multidimensional Functions. Information 2022, 13, 217.
44. Na, S., Xumin, L., & Yong, G. (2010, April). Research on k-means clustering algorithm: An improved k-means clustering algorithm. In 2010 Third International Symposium on intelligent information technology and security informatics (pp. 63-67). Ieee.
45. Nazeer, K. A., & Sebastian, M. P. (2009, July). Improving the Accuracy and Efficiency of the k-means Clustering Algorithm. In Proceedings of the world congress on engineering (Vol. 1, pp. 1-3). London, UK: Association of Engineers.
46. Ali, H. H., & Kadhum, L. E. (2017). K-means clustering algorithm applications in data mining and pattern recognition. International Journal of Science and Research (IJSR), 6(8), 1577-1584.
47. Ahmed, M., Seraj, R., & Islam, S. M. S. (2020). The k-means algorithm: A comprehensive survey and performance evaluation. Electronics, 9(8), 1295.
48. Hassanien, A. E., & Emary, E. (2018). Swarm intelligence: principles, advances, and applications. CRC press.
49. Tang, J., Liu, G., & Pan, Q. (2021). A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends. IEEE/CAA Journal of Automatica Sinica, 8(10), 1627-1643.
50. Floudas, C. A., & Pardalos, P. M. (Eds.). (2013). State of the art in global optimization: computational methods and applications
51. Horst, R., & Pardalos, P. M. (Eds.). (2013). Handbook of global optimization (Vol. 2). Springer Science & Business Media.
52. Cánovas, M. J., Kruger, A., Phu, H. X., & Théra, M. (2020). Marco A. López, a Pioneer of Continuous Optimization in Spain. Vietnam Journal of Mathematics, 48, 211-219.
53. Mahmoodabadi, M. J., & Nemati, A. R. (2016). A novel adaptive genetic algorithm for global optimization of mathematical test functions and real-world problems. Engineering Science and Technology, an International Journal, 19(4), 2002-2021.
54. Li, J., Xiao, X., Boukouvala, F., Floudas, C. A., Zhao, B., Du, G., ... & Liu, H. (2016). Data-driven mathematical modeling and global optimization framework for entire petrochemical planning operations. AIChE Journal, 62(9), 3020-3040.
55. Brezočnik, L., Fister Jr, I., & Podgorelec, V. (2018). Swarm intelligence algorithms for feature selection: a review. Applied Sciences, 8(9), 1521.
56. Houssein, E. H., Hosney, M. E., Mohamed, W. M., Ali, A. A., & Younis, E. M. (2023). Fuzzy-based hunger games search algorithm for global optimization and feature selection using medical data. Neural Computing and Applications, 35(7), 5251-5275.
57. Nadimi-Shahraki, M. H., Taghian, S., Mirjalili, S., Abualigah, L., Abd Elaziz, M., & Oliva, D. (2021). EWOA-OPF: Effective whale optimization algorithm to solve optimal power flow problem. Electronics, 10(23), 2975.
58. Boursianis, A. D., Papadopoulou, M. S., Salucci, M., Polo, A., Sarigiannidis, P., Psannis, K., ... & Goudos, S. K. (2021). Emerging swarm intelligence algorithms and their applications in antenna design: the GWO, WOA, and SSA optimizers. Applied Sciences, 11(18), 8330.
59. Zhang, J., & Wang, J. S. (2020). Improved whale optimization algorithm based on nonlinear adaptive weight and golden sine operator. IEEE Access, 8, 77013-77048.
60. Wan, Y., Mao, M., Zhou, L., Zhang, Q., Xi, X., & Zheng, C. (2019). A novel nature-inspired maximum power point tracking (MPPT) controller based on SSA-GWO algorithm for partially shaded photovoltaic systems. Electronics, 8(6), 680.
61. Ion, I. G., Bontinck, Z., Loukrezis, D., Römer, U., Lass, O., Ulbrich, S., ... & De Gersem, H. (2018). Robust shape optimization of electric devices based on deterministic optimization methods and finite-element analysis with affine parametrization and design elements. Electrical Engineering, 100(4), 2635-2647.
62. Cuevas-Velásquez, V., Sordo-Ward, A., García-Palacios, J. H., Bianucci, P., & Garrote, L. (2020). Probabilistic model for real-time flood operation of a dam based on a deterministic optimization model. Water, 12(11), 3206.
63. Pereyra, M., Schniter, P., Chouzenoux, E., Pesquet, J. C., Tourneret, J. Y., Hero, A. O., & McLaughlin, S. (2015). A survey of stochastic simulation and optimization methods in signal processing. IEEE Journal of Selected Topics in Signal Processing, 10(2), 224-241.
64. Hannah, L. A. (2015). Stochastic optimization. International Encyclopedia of the Social & Behavioral Sciences, 2, 473-481.
65. Kizielewicz, B., & Sałabun, W. (2020). A new approach to identifying a multi-criteria decision model based on stochastic optimization techniques. Symmetry, 12(9), 1551.

66. Chen, T., Sun, Y., & Yin, W. (2021). Solving stochastic compositional optimization is nearly as easy as solving stochastic optimization. IEEE Transactions on Signal Processing, 69, 4937-4948.
67. Gad, A. G. (2022). Particle swarm optimization algorithm and its applications: a systematic review. Archives of computational methods in engineering, 29(5), 2531-2561.
68. Zivkovic, M., Stoean, C., Chhabra, A., Budimirovic, N., Petrovic, A., & Bacanin, N. (2022). Novel improved salp swarm algorithm: An application for feature selection. Sensors, 22(5), 1711.
69. Emambocus, B. A. S., Jasser, M. B., & Amphawan, A. (2023). A survey on the optimization of artificial neural networks using swarm intelligence algorithms. IEEE Access, 11, 1280-1294.
70. Xu, M., Cao, L., Lu, D., Hu, Z., & Yue, Y. (2023). Application of swarm intelligence optimization algorithms in image processing: A comprehensive review of analysis, synthesis, and optimization. Biomimetics, 8(2), 235.