Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/mbe

MBE, 19(12): 14173–14211.
DOI: 10.3934/mbe.2022660
Received: 08 August 2022
Revised: 06 September 2022
Accepted: 08 September 2022
Published: 27 September 2022

Research article

Enhanced Aquila optimizer algorithm for global optimization and constrained engineering problems

Huangjing Yu[1,*], Heming Jia[1,*], Jianping Zhou[1] and Abdelazim G. Hussien[2,3,*]

[1] School of Information Engineering, Sanming University, Sanming 365004, China
[2] Department of Computer and Information Science, Linköping University, 581 83 Linköping, Sweden
[3] Faculty of Science, Fayoum University, Faiyum 63514, Egypt

* **Correspondence:** Email: 19990445@fjsmu.edu.cn, jiaheming@fjsmu.edu.cn, abdelazim.hussien@liu.se.

Abstract: The Aquila optimizer (AO) is a recently developed swarm algorithm that simulates the hunting behavior of Aquila birds. In complex optimization problems, an AO may have slow convergence or fall in sub-optimal regions, especially in high complex ones. This paper tries to overcome these problems by using three different strategies: restart strategy, opposition-based learning and chaotic local search. The developed algorithm named as mAO was tested using 29 CEC 2017 functions and five different engineering constrained problems. The results prove the superiority and efficiency of mAO in solving many optimization issues.

Keywords: Aquila optimizer; AO; restart strategy; opposition-based; chaotic local search

1. Introduction

The process of selecting appropriate and applicable variable values for a specific task is known as optimization [1–3]. Optimization exists in almost every domain, including job shop scheduling [4], feature selection [5–7], image processing [8,9], face detection and recognition [10], predicting chemical activities [11], classification [12,13], network allocation [14], internet of vehicles [15], routing [16], and neural network [17]. Due to the nature of real-world problems, Optimization becomes very challenging and has many difficulties such as multiobjectivity [18], memetic optimization [19], large-scale optimization [20], fuzzy optimization [21], uncertainties [22] and parameter estimation [23]. Meta-heuristics algorithms have been used to solve such problems due to their advantages such as flexibility, efficiency and getting a near-optimal solution in a reasonable time.

Table 1. Summary of literature review on Aquila optimizer (AO) variants and applications.

SN.	Modification Name	Authors	Remarks
1	Original AO	Abualigah et al. [49]	Authors simulate Aquila behavior
2	IHAOHHO	Wang et al. [50]	Hybrid Aquila optimizer and Harris hawks algorithm
3	Simplified AO (IAO)	Zhao et al. [54]	Only two techniques are used and others were dropped
4	AGWO	Ma et al. [55]	GWO is hybridized with AO
5	Aqu	Abdelaziz et al. [58]	COVID-19 images

Examples of metaheuristics algorithms include particle swarm optimization (PSO) [24], artificial bee colony [25], coot bird [26], genetic algorithms (GAs) [27], the krill herd algorithm [28], the harmony search (HS) algorithm [29], the snake optimizer [30], monarch butterfly optimization [31], the slime mold algorithm [32], the moth search algorithm [33], the hunger games search [34], the Runge-Kutta method [35], the weighted mean of vectors [36], the virus colony search [37], the lightning search algorithm [38], ant lion optimization [39], the crow search algorithm [40], moth-flame optimization [41], the wild horse optimizer [42], the remora optimization algorithm [43], the artificial rabbit Optimizer [44], the artificial hummingbird algorithm [45], grasshopper optimization algorithm (GOA) [46], grey wolf optimizer (GWO) [47] and the whale optimization algorithm (WOA) [48].

The Aquila optimizer (AO) is the latest developed algorithm proposed by Abualigah et al. [49] which simulates the four different phases of Aquila hunting behavior. Wang et al. [50] developed an improved version of the AO by replacing AO's original exploitation phase with the Harris hawks optimizer's exploitation phase. Moreover, they embedded a random opposition-learning strategy and nonlinear escaping operator in their proposed algorithm. They argued that the proposed algorithm is able to achieve the best results compared with the other five metaheuristic optimizers. Also, Mahajan et al. [51] hybridized the AO with the arithmetic optimization algorithm (AOA) [52]. They tested their algorithm which is called AO-AOA with original AO, original AOA, WOA, GOA and GWO. Another hybrid work between AO and AOA has been done by Zhang et al. [53]. Likewise, Zhao et al. [54] developed another version of the AO called the simplified AO algorithm by removing the control equation of the exploitation and exploration procedures (latter strategies) and keeping the former two techniques. They said their developed algorithm IAO achieved better results than many newly developed swarm algorithms. Another enhancement has been done by Ma et al. [55] in which grey wolf optimizer is hybridized using Aquila algorithm that allows some wolves to be able to fly, improving their search techniques, and avoiding getting stuck in local optima. They tested their developed algorithm with many optimizers using 23 functions. Also, Gao et al. [56] employed three different strategies to enhance the AO algorithm. These strategies are Gaussian mutation (GM), random-opposition learning, and developing a search control operator. They argued that their algorithm, an improved AO, has superior results compared to other optimizers.

The AO has been successfully used in many applications. For example, AlRassas et al. [57] tried to forecast oil production by using the AO to optimize the adaptive neuro-fuzzy inference system model. Also, Abdelaziz et al. [58] tried to classify COVID-19 images using the AO algorithm and MobilNet3. Likewise, Fatani et al. [59] developed an extraction and selecting approach for features using the AO and deep learning for iot detection intrusion systems.

Despite the powerfulness and superiority of the algorithm, and as stated by the no free lunch theorem, the AO cannot solve all optimization issues. So, the AO still needs more enhancements and developments.

This paper introduces a novel version of the AO in which three different strategies have been used to overcome the original optimizer drawbacks such as getting stuck in local optima and slow convergence. These strategies are the chaotic local search (CLS), opposition-based learning (OBL) and the restart strategy (RS). Using OBL and the RS enhances the AO exploratory search capabilities whereas the CLS improves AO exploitative search abilities.

The main contributions of this paper are as follows:

- A novel Aquila algorithm has been developed using three strategies: OBL, the RS and the CLS.
- The developed optimizer has been compared with the original AO and nine other algorithms, namely, the CSA [40], EHO [60], GOA [46], LSHADE [61], Lshade-EpSin [62], MFO [63], MVO [64], and PSO [24].
- A scalability test and removing one strategy from the developed algorithm experiments have been carried out.
- mAO was tested using 29 functions and five constrained ones.

This paper is organized as follows: Section 2 discusses the background and preliminaries of the original algorithm, OBL, the CLS and the RS, whereas Section 3 introduces the structure of the modified optimizer and its complexity. Sections 4 and 5 discuss the results of the proposed mAO and other competitors in CEC2017 and five different constrained engineering problems whereas Section 6 concludes the paper.

2. Preliminaries

2.1. Aquila optimizer (AO)

Aquila algorithm is one of the latest population-based swarm intelligence optimizers developed by Abualigah et al. [49]. Aquila can be considered among the most well-known prey birds existed in north hemisphere. Aquila is brown with a golden back body. Aquila uses its agility and strength with its wide nails and strong feet to catch various types ofprey usually squirrels, rabbits, marmots, and hares [65].

Aquila optimizer (AO) simulates the four different Aquila strategies in hunting. The next subsection shows Aquila's mathematical model.

2.2. Mathematical model

AO begins with a random set of individuals that can be represented mathematically as follows:

$$X = \begin{bmatrix} X_{1,1} & X_{1,2} & ... & X_{1,j} & ... & X_{1,D-1} & X_{1,D} \\ X_{2,1} & X_{2,2} & ... & X_{2,j} & ... & X_{2,D-1} & X_{2,D} \\ ... & ... & ... & ... & ... & & \\ X_{n-1,1} & X_{n-1,2} & ... & X_{n-1,j} & ... & X_{n-1,D-1} & X_{n-1,D} \\ X_{n,1} & X_{n,2} & ... & X_{n,j} & ... & X_{n,D-1} & X_{n,D} \end{bmatrix} \tag{2.1}$$

where X is an agent position (solution) that can be computed using the following equation:

$$X_{i,j} = rand \times (UB_j - LB_j) + LB_j, \quad i = 1, ..., n, j = 1, ..., D \tag{2.2}$$

where D refers to the number of decision variables, N indicates the number of agents, UB_j and LB_j are the j^{th} upper and lower boundaries, and x_i refers to the i^{th} value of the decision variable.

AO simulates Aquila hunting in four different phases where the optimizer can easily move from exploration and exploitation using the following condition:

$$\text{IF,} \begin{cases} t \leq (\frac{2}{3}) \times T & \text{Perform exploration} \\ \text{Otherwise} & \text{Perform exploration} \end{cases}$$

2.2.1. Expanded exploration (X_1)

In this phase, Aquila will determine the area to hunt the prey and select it by a vertical stoop and high soar. The mathematical formula for such a behavior is given by the following two equations:

$$X_1(t+1) = X_{best}(t) \times (1 - \frac{t}{T}) + (X_M(t) - X_{best}(t) * rand) \tag{2.3}$$

$$X_M(t) = \frac{1}{N} \sum_{i=1}^{N} X_i(t), \forall j = 1, 2, ..., Dim \tag{2.4}$$

where $X_M(t)$ indicates the mean position in the ith generation, X_{best} is the best Aquila position founded in this iteration, r is a randomly generated number in the interval $[0, 1]$, t is the current generation where T is the maximum generation number, and N is the Aquilas number.

2.2.2. Narrowed exploration (X_2)

This technique is the most technique used by Aquila for hunting. To attack the prey, a short gliding is used with contour flight. Aquila position's will be updated as follows:

$$X_2(t+1) = X_{best}(t) \times Levy(D) + X_R(t) + (y - x) * rand \tag{2.5}$$

where X_R indicates a position of Aquila generated randomly, *rand* is a random real number between 0 and 1, D is the number of variables, and *Levy* refers to a lévy function which is presented as follows:

$$Levy(D) = s \times \frac{u \times \sigma}{|\nu|^{\frac{1}{\beta}}} \tag{2.6}$$

$$\sigma = \frac{\Gamma(1+\beta) \times sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{\frac{\beta-1}{2}}} \tag{2.7}$$

where s is a fixed value and equals 0.01, u and μ are random numbers between 0 and 1 and β is a constant and equals 1.5. Both y and x are used to model the spiral shape and can be computed using the following two equations:

$$y = r \times \cos(\theta) \tag{2.8}$$

$$x = r \times \sin(\theta) \tag{2.9}$$

where r and θ can be calculated as follows:

$$r = r_1 + U \times D_1 \tag{2.10}$$

$$\theta = -\omega \times D_1 + \theta_1 \tag{2.11}$$

$$\theta_1 = \frac{3 \times \pi}{2} \tag{2.12}$$

where U equals 0.00565, ω equals 0.005, and r_1 has a value between 1 and 20.

2.2.3. Expanded exploitation (X_3)

In the 3^{rd} technique, the prey area is determined and agents can vertically perform a preliminary attack with low flight. Agents can attack the prey as follow:

$$X_3(t+1) = (X_{best}(t) - X_M(t)) \times \alpha - rand + ((UB - LB) \times rand + LB) \times \delta \tag{2.13}$$

where $X_M(t)$ indicates the mean position in the i-th generation, X_{best} is the best Aquila position founded in this iteration, $rand$ is a randomly generated number in the interval $[0, 1]$, α and β are exploitation parameters that are equal 0.1, and UB and LB refer to the upper and lower boundaries.

2.2.4. Narrowed exploitation (X_4)

In this phase, Aquila can easily chase the prey and attack attacks it using escape trajectory light which can be modeled as follows:

$$X_4(t+1) = QF \times X_{best}(t) - (G_1 \times X(t) \times rand) - G_2 \times Levy(D) + rand \times G_1 \tag{2.14}$$

$$QF(t) = t^{\frac{2 \times rand - 1}{(1-r)^2}} \tag{2.15}$$

$$G_1 = 2 \times rand - 1 \tag{2.16}$$

$$G_2 = 2 \times (1 - \frac{t}{T}) \tag{2.17}$$

where $QF(t)$ is the quality value, G_1 refers to various AO motions, and G_2 refers to chasing prey flight slope.

2.3. Opposition-based learning

Opposition-based learning strategy is a technique introduced by Tizhoosh [66] which has been employed by many researchers to improve many swarm optimizers. For example, Hussien [67] embedded OBL in SSA to overcome getting trapped in local optima. Moreover, Hussien and Amin used OBL with chaotic local search to improve the exploration abilities of HHO [7]. Zhao et al. employed OBL with arithmetic optimization algorithm [1]. OBL works by comparing the original solution with its opposite one. Let x is a real number that falls in the interval $[lb, ub]$, then its opposite can be calculated from the following equation:

$$\bar{x} = ub + lb - x \tag{2.18}$$

where *lb* and *ub* are lower boundary and upper one respectively, and $\bar{x}$ indicates the opposite solution. If x is a vector that has multi values, then $\bar{x}$ can be computed from the following equation:

$$\bar{x}_j = ub_j + lb_j - x_j \tag{2.19}$$

where x_j indicates the j^{th} value of x and ub_j and lb_j refer to upper and lower boundaries respectively.

2.4. Chaotic local search

The chaotic local search (CLS) technique has been integrated with many swarm optimizers such as WOA [68], HHO [7], brain storm optimization [69], and Jaya Algorithm (JAYA) [70]. CLS technique is almost used with the logistic map which is given in the following equation:

$$o^{s+1} = Co^s(1 - o^s) \tag{2.20}$$

where s is the number of the current iteration, C is a control parameter that equals 4, $o^1 \neq 0.25$, 0.50 and 0.75. Local search is used to search in the neighborhood area of the already founded optimal solution. CLS can be represented by the following equation:

$$Cs = (1 - \mu) \times T + \mu \grave{C}_i, i = 1, 2, ..., n \tag{2.21}$$

where Cs refers to the value generated by CLS in iteration i and $\grave{C}_i$ can be easily calculated as follows:

$$\grave{C}_i = LB + C_i \times (UB - LB) \tag{2.22}$$

μ is a shrinking factor and can be computed from the following below equation:

$$\mu = \frac{T - t + 1}{T} \tag{2.23}$$

where t and T refer to the current and maximum number of iterations.

2.5. Restart strategy

During the search operation, some agents may not be able to find a better location as they may get trapped in local optimum regions. Such agents may affect the overall search as they take many generation resources and don't enhance the search process. Restart strategy (RS) at this point which is proposed by Zhang et al. [71] can help worse agents to jump out from local regions. RS counts the number of times for each individual that has been enhanced and updated. So, if the i-th agent has updated, then the trial value will be zero, otherwise, the trial value will be increased by 1. If the trial is equal to a certain threshold, then the individual position will be changed using the following 2 equations:

$$X(t + 1) = lb + rand.(ub - lb) \tag{2.24}$$

$$X(t + 1) = rand.(ub + lb) - X(t) \tag{2.25}$$

where *ub* and *lb* refer to upper and lower boundaries and *rand* indicates a random number in the number $[0, 1]$.

3. Enhanced Aquila optimizer

3.1. Shortcoming of Aquila algorithm

AO similar to other swarm optimizers may get stagnation in sub-optimal areas and have a slow convergence, especially when addressing and handling complicated & complex problems that have high dimensional features.

3.2. Architecture of modified AO

Our proposed algorithm which is termed mAO tries to solve the original optimizer limitations. In the proposed mAO, three different strategies are used to improve the classical AO namely: opposition-based Learning, restart strategy, and chaotic local search. OBL strategy is used in both the initialization phase and updating agent position process. OBL is used in initialization by selecting the best N solutions from the pool of $X \cup \bar{x}$ to ensure the algorithm starts with a good set of agents whereas it is embedded in the updating process to improve algorithm exploration abilities. Moreover, a chaotic local search mechanism is used to improve the best solution and existed until now which will lead to the enhancement of the whole individuals. On the other hand, a restart strategy is employed in AO to change the position of the worst individuals if they have already get fallen in local regions. The pseudo-code of the developed optimizer can be seen in algorithm 1.

3.3. Complexity of mAO

The complexity of the proposed algorithm can be computed by calculating the complexity of each phase separately, i.e., initialization, evaluation, and updating process. So $O(\text{mAO}) = O(\text{Initialization}) + O(\text{Evaluation}) + O(\text{Updating Position}) + O(\text{CLS + OBL + RS})$. If D is the number of dimensions, N is the number of individuals, and T is the max iteration number, the following can be obtained.
$O(\text{Initialization}) = O(N)$
$O(\text{Evaluation}) = O(N \times T)$
$O(\text{Updating Position}) = O(N \times T \times D)$
$O(\text{CLS}) = O(N \times T)$
$O(\text{OBL}) = O(N \times T \times D)$
$O(\text{RS}) = O(N \times D)$
$O(\text{CLS + OBL + RS}) = O(N \times T \times D)$
$O(\text{mAO}) = O(N) + O(N \times T) + O(N \times T \times D) + O(N \times T \times D) = O(N \times T \times D)$

4. Experiments and discussion

4.1. Parameter setting

To validate our proposed approach, 29 functions from CEC2017 have been used to test mAO performance. These CEC2017 functions are very challenging and contain different types of functions (Unimodal, multimodal, composite, and hybrid). The description of CEC2017 functions is shown in Table 4 where *opt*. refers to the global optimal value. All experiments have been performed on Matlab 2021b using Intel Corei7 and 8.00 G of RAM. The parameter setting of all experiments is shown in Table 2. mAO is compared with the original Aquila Optimizer and other nine well-known

Algorithm 1 Improved Aquila optimizer

```
1: Initialize the population $X$ of the AO
2: Calculate $\overline{X}$ and select the best $N$ from ($X$ U $\overline{X}$)
3: Initialize AO parameters
4: while ($t < T$) do
5:     Compute the objective function values
6:     Select the best agent $X_{best}$
7:     for ($i = 1, 2, ..., N$) do
8:         Update the current solution mean
9:         Compute $y$, $x$, $G_1, G_2$ and $Levy(D)$
10:        if ($t \leq (\frac{2}{3})T$) then
11:            if $rand \leq 0.5$ then
12:                Update current position using Eq (2.3)
13:                Compute opposite position using Eq (2.19)
14:            else
15:                Update current position using Eq (2.5)
16:                Compute opposite position using Eq (2.19)
17:                if Fitness ($rand \leq 0.5$) then
18:                    Update the current solution using Eq (2.13)
19:                    Compute opposite position using Eq (2.19)
20:                else
21:                    Update the current solution using Eq (2.14)
22:                    Compute opposite position using Eq (2.19)
23:                end if
24:            end if
25:        end if
26:    end for
27:    Apply RS strategy using Eqs (2.24) and (2.25)
28:    Apply CLS strategy using Eq (2.21)
29: end while
30: Return best solution
```

Table 2. Experiments parameters settings.

No.	Parameter Name	Value
1	Population Size	30
2	Dim	30
3	Max number of iteration	500

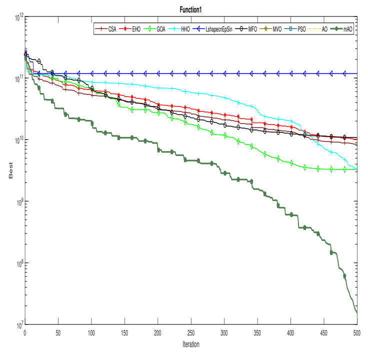
and powerful swarm algorithms namely: crow search algorithm [40], elephant herd optimizer [60], grasshopper optimization algorithm [46], LSHADE [61], Lshade-EpSin [62], moth-flame optimization [63], multi-verse optimization [64], and particle swarm algorithm [24]. The parameter of each mentioned algorithm is given in Table 3.

4.2. CEC2017

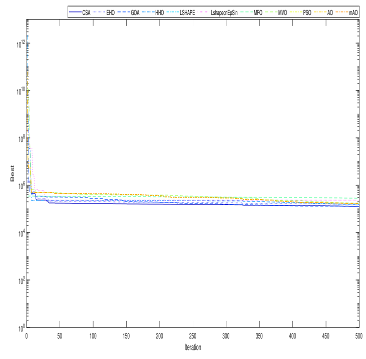
The developed optimizer and its competitors' results are shown in Table 5 in terms of best (min), worst (mac), mean (average), and standard deviation. From the above-mentioned table, it can be seen that the suggested technique has good results and performs well in solving all functions type. For example, in term of average, it ranked first in all unimodal functions ($F1$ and $F3$), and all multimodal functions ($F4$ and $F10$). On the other hand, it can be noticed that mAO achieved better results compared to the original optimizer and others. It ranked first in almost functions whereas it ranked first in solving composite problems in 5 functions out of 10. Besides the statistical measures, convergence curve can be seen as a powerful tool to compare any new algorithm with its competitors to see if it has a good convergence or slow one. mAO has been recognized to achieve a fast convergence curve in all mentioned function types as shown in Figures 1–3.

Furthemore, a statistical comparsion using Wilcoxon test [72, 73] has been carried out between the developed algorithm and all other competitors. Table 6 shows the p-values which show a big diffeence in the outputs between different optimizers. From Table 6, results prove the mAO algorithm superiority in finding near-optimal solutions when compared with others. To show the powerful and efficient of the proposed algorithm, a scalability test has been performed on 10 and 50 dimensions using the same functions and the same comparing algorithms. The results of this scalability test are shown in Table7. It can be seen that mAO is better than other competitors in almost functions.

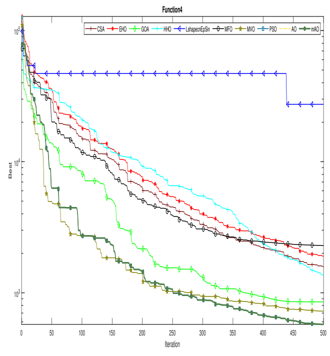
To show the powerfulness of our suggested optimizer by intergrating three strategies with AO, we test the standard AO with each operator seperately. Table 8 shows the average and standard deviation of four algorithms: AO with OBL (AOOBL), AO with CLS (AOCLS), AO with RS (AORS), and the developed algorithm mAO that contains AO with CLS, RS, and OBL.



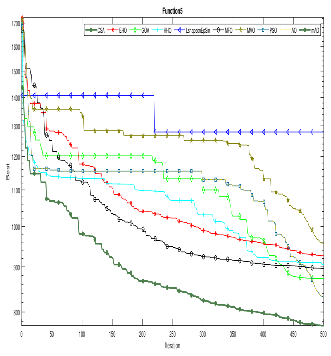
(a) F1



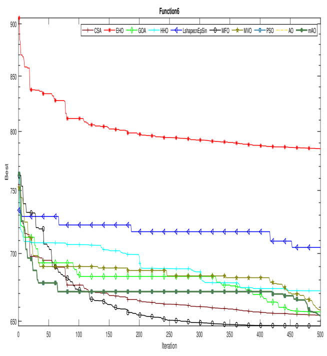
(b) F3



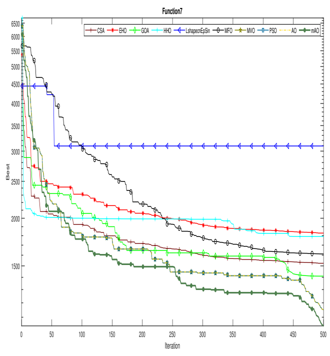
(c) F4



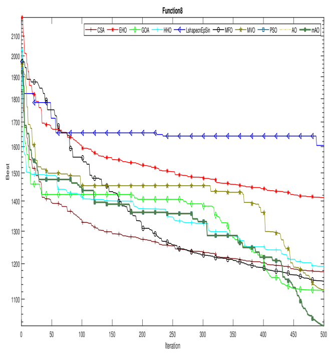
(d) F5



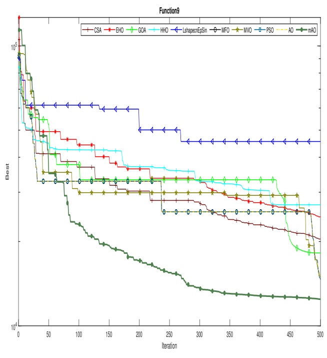
(e) F6



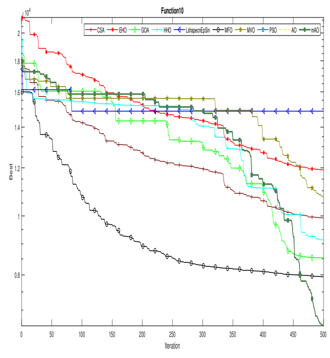
(f) F7



(g) F8

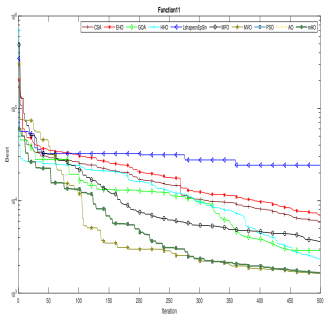


(h) F9

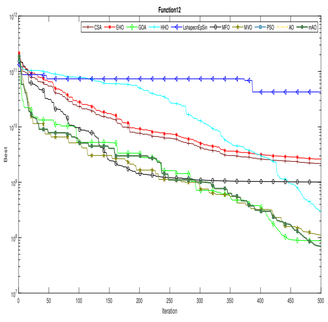


(i) F10

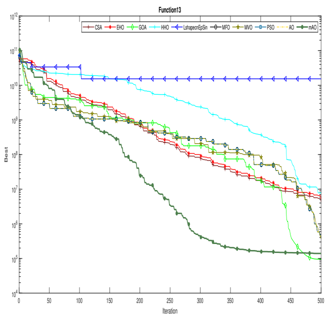
Figure 1. The convergence behavior of the comparative methods on CEC2017 F1–F10 functions, Dim = 30.



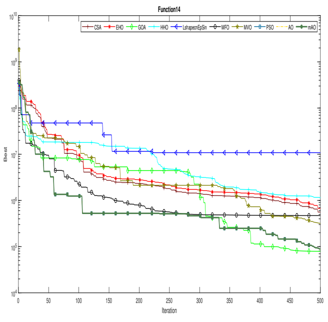
(**a**) F11



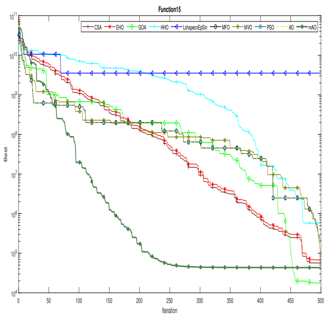
(**b**) F12



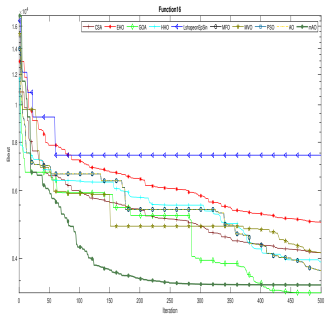
(**c**) F13



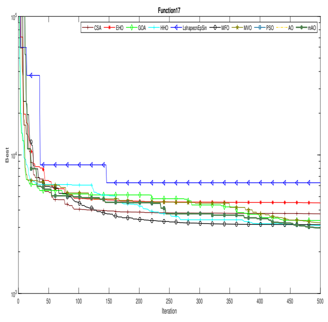
(**d**) F14



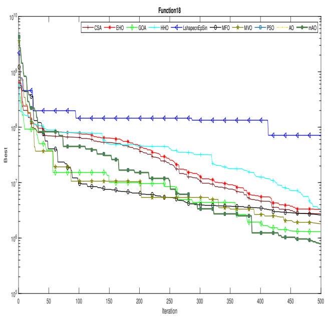
(**e**) F15



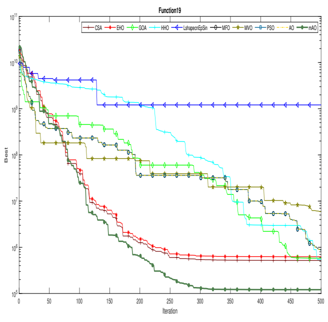
(**f**) F16



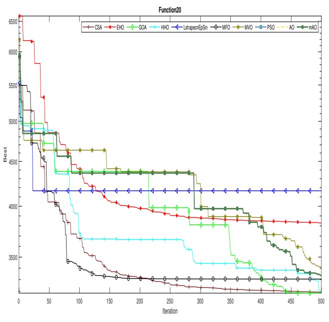
(**g**) F17



(**h**) F18

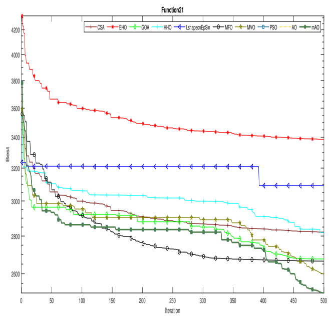


(**i**) F19

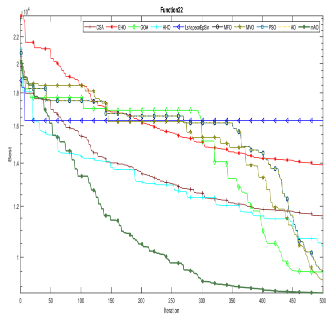


(**j**) F20

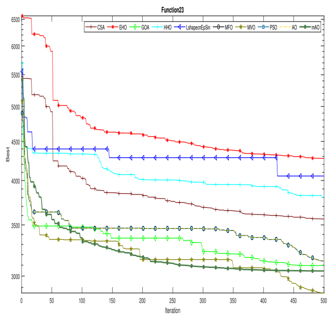
Figure 2. The convergence behavior of the comparative methods on CEC2017 F11–F20 functions, Dim = 30.



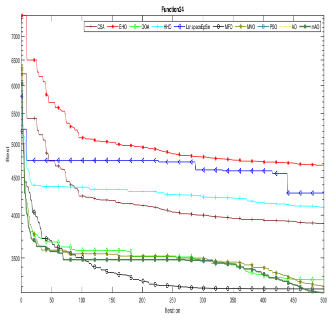
(a) F21



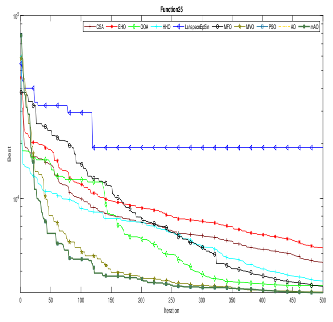
(b) F22



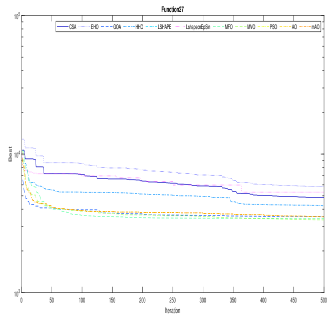
(c) 23



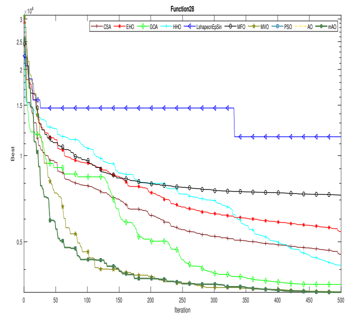
(d) F24



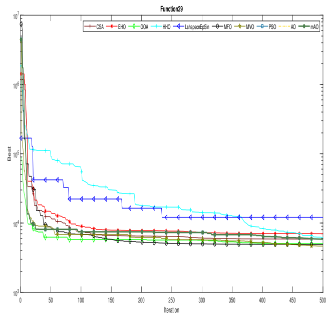
(e) F25



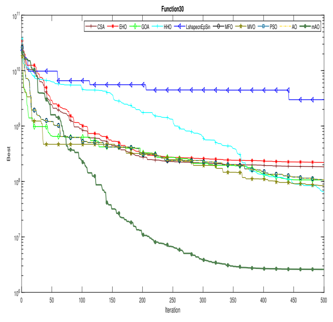
(f) F27



(g) F28



(h) F29



(i) F30

Figure 3. The convergence behavior of the comparative methods on CEC2017 F21–F30 functions, Dim = 30

Table 3. Setting of all meta-heuristic algorithms parameters.

Alg.	**Parameter**	**Value**
CSA	AP	0.1
	fl	2
EHO	$Numclanx$	5
	α	0.5
	β	0.1
	number of elite	2
GOA	c_{max}	1
	c_{min}	0.00004
L-SHADE	Pbest	0.1
	Arc rate	2
HHO	α	1.5
LSHADE-EpSin	H	5
	NPmin	4
	Pbest rate	0.11
	Arc rate	1.4
	ps	0.4
	pc	0.4
MFO	a	$\in [-2, 1]$
	Spiral factor b	1
MVO	p	6
PSO	wMax	0.9
	wMin	0.2
	c_1	2.0
	c_2	2.0
AO	U	0.00565
	r_1	10
	ω	0.005
	α	0.1
	δ	0.1
	G1	$\in [-1, 1]$
	G2	$\in [2, 0]$
mAO	U	0.00565
	r_1	10
	ω	0.005
	α	0.1
	δ	0.1
	G1	$\in [-1, 1]$
	G2	$\in [2, 0]$
	C	4

Table 4. Benchmark functions.

No.	Types	Name	Opt.
F1	Unimodal	Shifted and Rotated Bent Cigar Function	100
F2		Shifted and Rotated Sum of Different Power Function	200
F3		Shifted and Rotated Zakharov Function	300
F4	Multimodal	Shifted and Rotated Rosenbrock's Function	400
F5		Shifted and Rotated Rastrigin's Function	500
F6		Shifted and Rotated Expanded Scaffer's F6 Function	600
F7		Shifted and Rotated Lunacek Bi-Rastrigin Function	700
F8		Shifted and Rotated Non-Continuous Rastrigin's Function	800
F9		Shifted and Rotated Lévy Function	900
F10		Shifted and Rotated Schwefel's Function	1000
F11	Hybrid	H-Fun 1 (N = 3)	1100
F12		H-Fun 2 (N = 3)	1200
F13		H-Fun 3 (N = 3)	1300
F14		H-Fun 4 (N = 4)	1400
F15		H-Fun 5 (N = 4)	1500
F16		H-Fun 6 (N = 4)	1600
F17		H-Fun 6 (N = 5)	1700
F18		H-Fun 6 (N = 5)	1800
F19		H-Fun 6 (N = 5)	1900
F20		H-Fun 6 (N = 6)	2000
F21	Composition	C-Fun 1 (N = 3)	2100
F22		C-Fun 2 (N = 3)	2200
F23		C-Fun 3 (N = 4)	2300
F24		C-Fun 4 (N = 4)	2400
F25		C-Fun 5 (N = 5)	2500
F26		C-Fun 6 (N = 5)	2600
F27		C-Fun 7 (N = 6)	2700
F28		C-Fun 8 (N = 6)	2800
F29		C-Fun 9 (N = 3)	2900
F30		C-Fun 10 (N = 3)	3000

Table 5. The comparison results of all algorithms over 30 functions.

F		CSA	EHO	GOA	HHO	LSHADE	Lshade-EpSin	MFO	MVO	PSO	AO	mAO
F1	Best	4.31E+09	5.06E+09	3.06E+09	2.09E+09	3.51E+10	1.31E+11	2.08E+10	2.25E+10	2.88E+08	4.18E+08	1.50E+07
	Worest	1.08E+10	1.12E+10	9.40E+09	9.57E+09	6.22E+10	1.49E+11	9.10E+10	4.27E+10	2.76E+09	9.96E+08	2.62E+07
	Mean	7.39E+09	8.31E+09	5.64E+09	5.64E+09	4.05E+10	1.36E+11	3.80E+10	2.79E+10	5.49E+08	5.45E+08	**1.78E+07**
	Std	1.71E+09	1.68E+09	1.98E+09	1.96E+09	8.34E+09	9.47E+09	2.23E+10	7.22E+09	5.47E+08	2.02E+08	4.23E+06
F3	Best	1.27E+05	5.90E+09	1.20E+05	1.40E+05	2.62E+05	2.54E+05	3.04E+05	1.41E+05	2.41E+05	1.43E+05	9.90E+04
	Worest	1.79E+05	1.88E+10	4.65E+05	2.11E+05	4.78E+05	3.29E+05	5.58E+05	2.50E+05	3.63E+05	2.30E+05	1.77E+05
	Mean	1.32E+05	8.74E+09	2.18E+05	1.80E+05	3.38E+05	2.75E+05	3.92E+05	1.75E+05	2.71E+05	1.58E+05	**1.31E+05**
	Std	1.57E+04	2.92E+09	8.25E+04	1.86E+04	7.52E+04	3.57E+04	8.69E+04	4.36E+04	4.13E+04	2.55E+04	1.55E+04
F4	Best	1.35E+03	5.25E+09	8.43E+02	1.23E+03	6.79E+03	3.72E+04	2.72E+03	4.02E+03	7.90E+02	7.75E+02	5.35E+02
	Worest	2.93E+03	1.22E+10	1.68E+03	3.48E+03	1.27E+04	4.67E+04	6.81E+03	9.14E+03	1.19E+03	1.05E+03	7.24E+02
	Mean	2.02E+03	9.03E+09	1.18E+03	1.91E+03	8.15E+03	3.92E+04	4.21E+03	5.24E+03	8.93E+02	8.36E+02	**5.91E+02**
	Std	4.89E+02	2.17E+09	2.63E+02	4.81E+02	2.40E+03	5.65E+03	1.49E+03	1.58E+03	1.21E+02	9.06E+01	6.13E+01
F5	Best	7.53E+02	5.93E+09	7.83E+02	8.92E+02	1.10E+03	1.32E+03	9.46E+02	1.06E+03	7.58E+02	8.18E+02	7.26E+02
	Worest	9.42E+02	1.64E+10	1.07E+03	1.06E+03	1.26E+03	1.41E+03	1.12E+03	1.14E+03	9.69E+02	9.36E+02	9.57E+02
	Mean	8.64E+02	9.26E+09	9.51E+02	9.45E+02	1.13E+03	1.36E+03	1.00E+03	1.08E+03	8.43E+02	8.64E+02	**7.77E+02**
	Std	4.91E+01	2.37E+09	6.86E+01	3.90E+01	6.41E+01	4.90E+01	7.50E+01	6.10E+01	7.57E+01	4.44E+01	2.55E+01
F6	Best	6.57E+02	4.24E+09	6.48E+02	6.66E+02	6.66E+02	7.07E+02	6.51E+02	6.45E+02	6.89E+02	6.53E+02	6.10E+02
	Worest	6.77E+02	1.51E+10	7.03E+02	6.87E+02	7.00E+02	7.18E+02	6.78E+02	6.92E+02	6.98E+02	6.85E+02	6.25E+02
	Mean	6.70E+02	9.19E+09	6.76E+02	6.79E+02	6.74E+02	7.11E+02	6.61E+02	6.53E+02	6.91E+02	6.60E+02	**6.15E+02**
	Std	4.73E+00	2.21E+09	1.36E+01	5.18E+00	9.81E+00	4.63E+00	1.07E+01	1.23E+01	4.54E+00	9.84E+00	3.37E+00

Continue on the next page

Table 5. The comparison results of all algorithms over 30 functions.

F		CSA	EHO	GOA	HHO	LSHADE	Lshade-EpSin	MFO	MVO	PSO	AO	mAO
F7	Best	1.40E+03	2.87E+09	1.41E+03	1.78E+03	1.83E+03	3.39E+03	1.71E+03	1.78E+03	1.24E+03	1.26E+03	1.15E+03
	Worest	1.82E+03	1.43E+10	1.88E+03	1.97E+03	2.38E+03	3.93E+03	2.86E+03	1.95E+03	1.42E+03	1.48E+03	1.34E+03
	Mean	1.58E+03	9.15E+09	1.61E+03	1.88E+03	1.95E+03	3.58E+03	2.02E+03	1.86E+03	1.27E+03	1.32E+03	**1.18E+03**
	Std	1.22E+02	2.93E+09	1.38E+02	6.27E+01	1.77E+02	2.55E+02	3.84E+02	7.46E+01	7.74E+01	7.97E+01	7.24E+01
F8	Best	1.10E+03	4.48E+09	1.09E+03	1.17E+03	1.39E+03	1.63E+03	1.25E+03	1.38E+03	1.09E+03	1.12E+03	1.06E+03
	Worest	1.28E+03	1.09E+10	1.33E+03	1.31E+03	1.51E+03	1.71E+03	1.48E+03	1.44E+03	1.22E+03	1.30E+03	1.17E+03
	Mean	1.20E+03	8.09E+09	1.23E+03	1.24E+03	1.44E+03	1.65E+03	1.29E+03	1.40E+03	1.13E+03	1.17E+03	**1.09E+03**
	Std	4.83E+01	2.11E+09	7.29E+01	3.56E+01	4.99E+01	3.18E+01	9.21E+01	2.62E+01	6.03E+01	6.46E+01	5.22E+01
F9	Best	9.40E+03	4.06E+09	1.60E+04	2.58E+04	2.70E+04	4.64E+04	1.78E+04	2.74E+04	4.18E+03	1.89E+04	1.49E+04
	Worest	2.12E+04	1.63E+10	3.73E+04	3.87E+04	4.56E+04	6.01E+04	3.99E+04	4.05E+04	4.32E+04	3.53E+04	3.15E+04
	Mean	1.54E+04	8.42E+09	2.66E+04	3.18E+04	3.27E+04	5.25E+04	3.11E+04	2.16E+04	8.35E+03	2.20E+04	**2.10E+04**
	Std	4.77E+03	2.73E+09	5.60E+03	3.69E+03	7.64E+03	5.09E+03	5.32E+03	8.22E+03	9.05E+03	6.91E+03	3.32E+03
F10	Best	7.57E+03	4.85E+09	8.21E+03	8.92E+03	1.56E+04	1.50E+04	8.49E+03	1.25E+04	1.40E+04	1.16E+04	7.05E+03
	Worest	1.46E+04	1.10E+10	1.21E+04	1.24E+04	1.72E+04	1.57E+04	1.10E+04	1.08E+04	1.58E+04	1.41E+04	1.05E+04
	Mean	9.31E+03	7.96E+09	1.02E+04	1.03E+04	1.62E+04	1.53E+04	9.24E+03	1.29E+04	1.45E+04	1.21E+04	**7.82E+03**
	Std	8.42E+02	1.92E+09	1.11E+03	8.32E+02	6.62E+02	7.99E+02	1.10E+03	1.06E+03	1.19E+03	1.42E+03	2.81E+02
F11	Best	2.63E+03	3.77E+09	2.68E+03	2.09E+03	1.82E+04	2.58E+04	6.14E+03	5.54E+03	1.84E+03	2.24E+03	1.58E+03
	Worest	6.26E+03	1.14E+10	9.92E+03	5.10E+03	4.98E+04	4.09E+04	3.48E+04	1.43E+04	2.78E+03	4.10E+03	1.97E+03
	Mean	4.52E+03	7.36E+09	4.76E+03	3.20E+03	2.53E+04	3.05E+04	1.63E+04	8.45E+03	2.10E+03	2.65E+03	**1.66E+03**
	Std	1.00E+03	S1.86E+09	1.96E+03	8.03E+02	1.01E+04	5.04E+03	9.70E+03	2.96E+03	3.18E+02	5.58E+02	1.19E+02
F12	Best	6.49E+08	4.25E+09	4.74E+07	2.19E+08	5.41E+09	5.12E+10	1.94E+09	6.06E+07	5.68E+09	3.34E+07	1.76E+07

Continue on the next page

Table 5. The comparison results of all algorithms over 30 functions.

F		CSA	EHO	GOA	HHO	LSHADE	Lshade-EpSin	MFO	MVO	PSO	AO	mAO
	Worest	2.19E+09	1.32E+10	1.02E+09	2.83E+09	1.25E+10	6.42E+10	1.05E+10	3.05E+08	1.98E+10	1.43E+08	1.46E+08
	Mean	1.26E+09	8.35E+09	3.63E+08	9.33E+08	6.99E+09	5.52E+10	3.89E+09	1.17E+08	9.44E+09	5.29E+07	**4.74E+07**
	Std	5.02E+08	2.11E+09	2.44E+08	6.13E+08	1.93E+09	7.75E+09	2.74E+09	6.99E+07	4.44E+09	3.05E+07	3.85E+07
F13	Best	1.49E+06	1.92E+09	6.11E+04	7.36E+06	9.12E+08	1.75E+10	7.50E+05	2.92E+05	1.14E+09	2.63E+04	1.24E+04
	Worest	6.05E+07	1.33E+10	1.37E+06	2.18E+08	2.41E+09	3.16E+10	6.52E+09	7.87E+05	6.61E+09	1.26E+05	2.98E+04
	Mean	1.40E+07	8.47E+09	4.43E+05	3.12E+07	1.55E+09	2.05E+10	1.33E+09	3.97E+05	2.08E+09	4.84E+04	**1.76E+04**
	Std	1.45E+07	2.77E+09	3.48E+05	4.58E+07	5.87E+08	4.92E+09	1.89E+09	1.52E+05	1.78E+09	2.80E+04	6.46E+03
F14	Best	1.06E+05	2.87E+09	7.01E+04	6.61E+05	2.39E+06	1.25E+07	8.65E+05	1.51E+05	1.14E+06	1.92E+05	4.49E+05
	Worest	2.65E+06	1.90E+10	2.48E+06	2.72E+07	2.52E+07	4.42E+07	1.83E+07	5.17E+05	1.28E+07	1.28E+06	1.78E+06
	Mean	9.35E+05	9.20E+09	6.14E+05	6.98E+06	6.92E+06	2.09E+07	3.35E+06	**2.16E+05**	7.00E+05	3.01E+06	4.70E+05
	Std	9.00E+05	3.45E+09	6.77E+05	8.03E+06	6.70E+06	9.15E+06	4.12E+06	1.07E+05	4.74E+05	2.91E+06	3.40E+05
F15	Best	1.72E+04	2.81E+09	1.57E+04	5.70E+05	2.06E+08	4.62E+09	5.41E+04	9.19E+04	1.25E+08	4.67E+03	3.34E+03
	Worest	1.77E+05	1.35E+10	4.60E+05	4.58E+07	1.07E+09	7.78E+09	4.18E+08	2.79E+05	2.68E+04	1.14E+09	2.39E+04
	Mean	5.53E+04	8.47E+09	1.09E+05	5.40E+06	3.74E+08	5.82E+09	6.28E+07	1.30E+05	9.53E+03	3.74E+08	**9.51E+03**
	Std	3.58E+04	2.44E+09	1.01E+05	1.06E+07	2.76E+08	1.54E+09	1.53E+08	5.82E+04	7.36E+03	3.64E+08	5.65E+03
F16	Best	3.49E+03	4.88E+09	3.24E+03	3.62E+03	6.09E+03	7.79E+03	3.53E+03	3.10E+03	3.62E+03	4.85E+03	3.35E+03
	Worest	6.54E+03	1.19E+10	5.05E+03	6.22E+03	7.18E+03	8.70E+03	5.51E+03	5.36E+03	5.31E+03	7.41E+03	4.21E+03
	Mean	4.79E+03	7.67E+09	4.14E+03	4.97E+03	6.42E+03	8.09E+03	4.20E+03	3.61E+03	4.19E+03	5.31E+03	**3.56E+03**
	Std	6.41E+02	1.91E+09	5.59E+02	7.09E+02	4.97E+02	4.27E+02	7.40E+02	6.06E+02	7.01E+02	7.31E+02	3.73E+02
F17	Best	2.92E+03	4.07E+09	2.88E+03	2.70E+03	4.79E+03	7.38E+03	3.70E+03	3.09E+03	3.32E+03	3.85E+03	2.89E+03
	Worest	3.85E+03	1.26E+10	4.41E+03	4.72E+03	5.52E+03	1.41E+04	5.32E+03	4.00E+03	4.75E+03	5.00E+03	3.94E+03

Continue on the next page

Table 5. The comparison results of all algorithms over 30 functions.

F		CSA	EHO	GOA	HHO	LSHADE	Lshade-EpSin	MFO	MVO	PSO	AO	mAO
	Mean	3.40E+03	7.99E+09	3.73E+03	3.70E+03	5.03E+03	8.94E+03	3.92E+03	3.31E+03	3.73E+03	4.24E+03	**3.23E+03**
	Std	2.67E+02	2.21E+09	3.70E+02	5.26E+02	2.96E+02	2.13E+03	5.22E+02	3.76E+02	6.02E+02	4.28E+02	4.27E+02
F18	Best	1.02E+06	4.40E+09	6.81E+06	3.22E+06	3.22E+07	9.35E+07	3.95E+06	7.89E+05	2.81E+06	1.73E+06	1.17E+06
	Worest	1.73E+07	1.10E+10	6.56E+07	3.75E+07	2.01E+08	1.90E+08	3.42E+07	9.04E+06	1.53E+07	7.00E+06	1.67E+06
	Mean	4.69E+06	8.10E+09	2.07E+07	1.00E+07	6.33E+07	1.12E+08	1.11E+07	**2.32E+06**	6.56E+06	3.13E+06	6.42E+06
	Std	4.24E+06	1.78E+09	1.80E+07	8.83E+06	4.68E+07	3.28E+07	8.56E+06	2.05E+06	4.19E+06	1.77E+06	4.56E+06
F19	Best	1.53E+05	5.36E+09	3.07E+05	2.23E+05	7.44E+07	1.67E+09	3.87E+05	2.23E+06	2.25E+03	5.96E+07	1.14E+04
	Worest	7.15E+06	1.27E+10	3.91E+07	1.42E+07	2.05E+08	3.63E+09	8.29E+08	1.62E+07	3.22E+04	3.87E+08	2.91E+04
	Mean	1.77E+06	9.34E+09	1.27E+07	3.84E+06	1.09E+08	2.33E+09	4.50E+07	5.65E+06	**8.79E+03**	1.21E+08	1.63E+04
	Std	1.64E+06	1.77E+09	9.06E+06	3.91E+06	4.59E+07	7.14E+08	1.85E+08	4.10E+06	1.02E+04	8.23E+07	7.65E+03
F20	Best	3.01E+03	4.10E+09	2.98E+03	3.15E+03	4.62E+03	4.29E+03	3.37E+03	3.12E+03	3.26E+03	3.12E+03	3.26E+03
	Worest	3.83E+03	1.67E+10	4.34E+03	3.89E+03	5.17E+03	4.63E+03	4.22E+03	4.11E+03	4.52E+03	3.93E+03	4.09E+03
	Mean	3.34E+03	8.95E+09	3.55E+03	3.52E+03	4.85E+03	4.35E+03	3.61E+03	**3.30E+03**	3.58E+03	3.38E+03	3.57E+03
	Std	2.28E+02	2.85E+09	2.95E+02	2.41E+02	2.25E+02	1.41E+02	2.53E+02	3.47E+02	5.21E+02	2.92E+02	3.06E+02
F21	Best	2.66E+03	3.63E+09	2.67E+03	2.79E+03	2.86E+03	3.14E+03	2.70E+03	2.51E+03	2.88E+03	2.56E+03	2.54E+03
	Worest	2.91E+03	1.45E+10	3.06E+03	3.36E+03	2.98E+03	3.22E+03	2.95E+03	2.63E+03	3.07E+03	2.80E+03	2.78E+03
	Mean	2.78E+03	9.28E+09	2.78E+03	2.96E+03	2.90E+03	3.17E+03	2.77E+03	**2.56E+03**	2.95E+03	2.66E+03	2.62E+03
	Std	6.26E+01	2.75E+09	9.73E+01	1.16E+02	4.33E+01	4.05E+01	8.87E+01	6.43E+01	9.74E+01	7.62E+01	5.30E+01
F22	Best	3.70E+03	3.89E+09	9.28E+03	1.02E+04	1.73E+04	1.66E+04	9.83E+03	1.49E+04	1.52E+04	1.29E+04	9.01E+03
	Worest	1.33E+04	1.74E+10	1.38E+04	1.48E+04	1.88E+04	1.76E+04	1.19E+04	1.75E+04	1.72E+04	1.57E+04	1.13E+04
	Mean	1.07E+04	9.31E+09	1.16E+04	1.21E+04	1.77E+04	1.69E+04	1.04E+04	1.56E+04	1.53E+04	1.30E+04	**9.65E+03**

Continue on the next page

Table 5. The comparison results of all algorithms over 30 functions.

F		CSA	EHO	GOA	HHO	LSHADE	Lshade-EpSin	MFO	MVO	PSO	AO	mAO
	Std	2.52E+03	3.40E+09	1.31E+03	1.12E+03	9.98E+02	4.37E+02	8.50E+02	1.43E+03	1.39E+03	3.81E+03	9.70E+02
F23	Best	3.46E+03	2.86E+09	3.09E+03	3.69E+03	3.43E+03	4.11E+03	3.11E+03	3.58E+03	3.03E+03	3.11E+03	2.96E+03
	Worest	4.09E+03	2.07E+10	3.64E+03	4.54E+03	3.61E+03	4.37E+03	3.38E+03	3.94E+03	3.27E+03	3.29E+03	3.21E+03
	Mean	3.83E+03	9.44E+09	3.34E+03	4.17E+03	3.49E+03	4.19E+03	3.16E+03	3.70E+03	3.11E+03	3.15E+03	**3.03E+03**
	Std	1.70E+02	3.78E+09	1.55E+02	2.43E+02	6.72E+01	9.66E+01	8.30E+01	1.42E+02	7.92E+01	6.70E+01	8.38E+01
F24	Best	3.61E+03	5.32E+09	3.22E+03	3.75E+03	3.59E+03	4.40E+03	3.67E+03	3.14E+03	3.30E+03	3.21E+03	3.19E+03
	Worest	4.59E+03	1.63E+10	3.73E+03	4.76E+03	3.88E+03	4.73E+03	3.88E+03	3.35E+03	3.44E+03	3.38E+03	3.36E+03
	Mean	4.05E+03	8.62E+09	3.45E+03	4.39E+03	3.65E+03	4.50E+03	3.73E+03	**3.18E+03**	3.33E+03	3.26E+03	3.25E+03
	Std	2.58E+02	2.54E+09	1.33E+02	2.47E+02	9.84E+01	1.46E+02	1.10E+02	7.17E+01	5.13E+01	6.32E+01	5.48E+01
F25	Best	3.69E+03	4.11E+09	3.26E+03	3.49E+03	7.53E+03	2.08E+04	3.76E+03	5.28E+03	3.23E+03	3.24E+03	3.05E+03
	Worest	4.49E+03	1.34E+10	4.29E+03	4.69E+03	1.11E+04	2.83E+04	1.32E+04	7.19E+03	3.43E+03	3.51E+03	3.13E+03
	Mean	4.04E+03	9.09E+09	3.65E+03	3.81E+03	8.59E+03	2.32E+04	5.33E+03	5.70E+03	3.30E+03	3.30E+03	**3.07E+03**
	Std	2.23E+02	2.52E+09	2.74E+02	2.92E+02	1.47E+03	2.85E+03	2.41E+03	7.05E+02	8.91E+01	7.88E+01	2.87E+01
F26	Best	3.69E+03	4.11E+09	3.26E+03	3.49E+03	7.53E+03	2.08E+04	3.76E+03	5.28E+03	3.23E+03	3.24E+03	3.05E+03
	Worest	4.49E+03	1.34E+10	4.29E+03	4.69E+03	1.11E+04	2.83E+04	1.32E+04	7.19E+03	3.43E+03	3.51E+03	3.13E+03
	Mean	4.04E+03	9.09E+09	3.65E+03	3.81E+03	8.59E+03	2.32E+04	5.33E+03	5.70E+03	3.30E+03	3.30E+03	**3.07E+03**
	Std	2.23E+02	2.52E+09	2.74E+02	2.92E+02	1.47E+03	2.85E+03	2.41E+03	7.05E+02	8.91E+01	7.88E+01	2.87E+01
F27	Best	4.20E+03	5.55E+09	3.51E+03	4.11E+03	4.14E+03	5.77E+03	3.54E+03	3.37E+03	3.59E+03	4.01E+03	3.55E+03
	Worest	6.55E+03	1.33E+10	4.39E+03	6.97E+03	4.81E+03	6.58E+03	3.79E+03	3.74E+03	3.88E+03	4.87E+03	4.07E+03
	Mean	5.42E+03	8.69E+09	3.78E+03	5.03E+03	4.30E+03	5.97E+03	3.62E+03	**3.47E+03**	3.66E+03	4.30E+03	3.68E+03

Continue on the next page

Table 5. The comparison results of all algorithms over 30 functions.

F		CSA	EHO	GOA	HHO	LSHADE	Lshade-EpSin	MFO	MVO	PSO	AO	mAO
	Std	6.27E+02	1.98E+09	2.42E+02	7.26E+02	.08E+02	3.70E+02	9.18E+01	1.26E+02	1.12E+02	2.89E+02	1.77E+02
F28	Best	4.29E+03	3.74E+09	3.55E+03	4.08E+03	7.96E+03	1.32E+04	7.92E+03	6.01E+03	3.48E+03	3.60E+03	3.31E+03
	Worest	6.13E+03	1.25E+10	4.74E+03	5.92E+03	9.92E+03	1.58E+04	1.04E+04	8.14E+03	4.38E+03	4.04E+03	3.38E+03
	Mean	4.88E+03	7.83E+09	3.95E+03	4.72E+03	8.11E+03	1.40E+04	8.57E+03	6.50E+03	3.59E+03	3.70E+03	**3.32E+03**
	Std	4.24E+02	2.00E+09	3.19E+02	4.47E+02	8.93E+02	1.25E+03	9.61E+02	7.38E+02	2.11E+02	1.33E+02	2.84E+01
F29	Best	5.53E+03	6.15E+09	5.01E+03	5.03E+03	6.91E+03	1.56E+04	4.97E+03	4.94E+03	4.29E+03	6.97E+03	4.62E+03
	Worest	8.87E+03	1.70E+10	6.87E+03	8.39E+03	1.02E+04	2.37E+04	6.81E+03	6.18E+03	5.63E+03	9.31E+03	5.87E+03
	Mean	7.19E+03	9.08E+09	5.85E+03	7.03E+03	7.81E+03	1.75E+04	5.49E+03	5.30E+03	**4.59E+03**	7.63E+03	5.05E+03
	Std	8.94E+02	2.80E+09	5.45E+02	8.24E+02	1.17E+03	3.26E+03	5.49E+02	4.35E+02	4.00E+02	8.04E+02	4.52E+02
F30	Best	8.44E+07	3.95E+09	7.63E+07	5.04E+07	4.05E+08	3.32E+09	9.10E+06	6.76E+07	5.88E+06	4.56E+08	3.43E+06
	Worest	3.37E+08	1.25E+10	2.69E+08	4.30E+08	8.01E+08	5.25E+09	1.37E+08	2.11E+08	7.29E+07	1.07E+09	9.93E+06
	Mean	2.25E+08	8.68E+09	1.63E+08	1.43E+08	4.71E+08	3.95E+09	3.56E+07	9.27E+07	1.24E+07	6.21E+08	**5.61E+06**
	Std	6.98E+07	2.24E+09	4.63E+07	8.02E+07	1.35E+08	7.58E+08	4.31E+07	3.74E+07	1.51E+07	2.22E+08	2.46E+06

Table 6. Wilcoxon rank sum test results for mAO against other algorithms CEC2017.

F	CSA	EHO	GOA	HHO	LSHADE	Lshade-EpSin	MFO	MVO	PSO	AO
F1	5.82E-08	3.77E-08	7.44E-08	9.09E-08	7.02E-02	8.06E-08	4.57E-08	6.75E-08	9.02E-08	7.32E-08
F3	2.83E-08	5.44E-08	7.90E-08	1.66E-07	1.12E-03	6.80E-08	7.80E-08	6.44E-08	3.55E-08	9.65E-08
F4	7.03E-08	5.87E-06	1.05E-06	6.80E-08	8.82E-01	6.85E-08	6.54E-08	4.82E-01	6.78E-08	6.83E-08
F5	9.74E-08	2.73E-03	1.26E-07	6.80E-07	5.33E-02	7.24E-04	8.87E-07	3.52E-03	5.50E-08	7.76E-07
F6	8.91E-07	2.81E-01	6.80E-07	6.77E-07	7.03E-06	5.80E-05	8.12E-04	4.37E-05	8.45E-07	7.75E-07
F7	3.22E-08	5.03E-04	7.72E-06	7.43E-08	3.56E-01	7.77E-08	6.44E-08	3.25E-04	6.66E-08	1.32E-08

Continue on the next page

Table 6. Wilcoxon rank sum test results for mAO against other algorithms CEC2017.

F	CSA	EHO	GOA	HHO	LSHADE	Lshade-EpSin	MFO	MVO	PSO	AO
F8	4.83E-08	7.23E-03	4.32E-06	7.22E-07	5.24E-03	6.82E-08	4.47E-08	4.52E-06	6.33E-07	4.81E-03
F9	7.88E-06	4.54E-06	3.31E-07	5.23E-08	6.42E-05	7.22E-08	1.09E-08	6.52E-05	7.42E-08	1.29E-08
F10	7.23E-08	7.67E-06	1.32E-07	6.53E-08	3.21E-07	6.94E-08	6.46E-08	3.31E-07	6.74E-08	6.65E-08
F11	5.77E-08	4.54E-05	9.86E-01	6.45E-08	4.75E-01	6.33E-08	6.23E-08	5.75E-01	4.33E-08	3.23E-08
F12	6.45E-08	3.18E-05	1.34E-07	6.84E-08	1.46E-04	6.83E-08	6.82E-08	1.46E-04	6.83E-08	6.82E-08
F13	6.70E-08	1.61E-04	6.23E-08	6.34E-08	4.45E-06	6.23E-08	6.87E-08	4.45E-06	6.23E-08	6.87E-08
F14	2.43E-03	3.53E-03	1.75E-03	7.06E-08	9.84E-03	6.14E-05	3.67E-06	9.64E-03	6.84E-05	3.47E-06
F15	6.54E-08	7.47E-06	7.75E-08	6.77E-08	1.34E-07	6.54E-08	6.23E-08	1.36E-07	6.58E-08	6.27E-08
F16	6.43E-08	2.22E-04	1.02E-06	6.86E-08	4.06E-03	6.43E-23	6.23E-08	4.11E-03	6.24E-08	6.31E-08
F17	3.46E-06	1.73E-03	3.97E-03	6.56E-08	5.53E-01	5.94E-04	6.76E-08	5.57E-01	5.96E-04	6.78E-08
F18	4.34E-06	9.75E-04	1.36E-03	3.84E-07	2.74E-02	3.88E-07	6.54E-08	2.64E-02	3.78E-07	6.34E-08
F19	6.83E-08	1.40E-04	6.60E-08	6.65E-08	6.63E-08	6.35E-08	6.76E-08	6.33E-08	6.97E-08	6.45E-08
F20	6.45E-08	4.93E-02	5.21E-03	1.43E-07	4.54E-01	2.42E-06	9.56E-08	4.22E-01	2.61E-06	9.65E-08
F21	6.46E-08	9.76E-05	7.65E-08	6.76E-08	8.48E-05	7.55E-08	6.43E-08	8.59E-05	7.65E-08	6.54E-08
F22	1.23E-03	2.74E-04	2.47E-07	6.45E-08	1.03E-06	8.85E-06	6.44E-08	1.06E-06	8.86E-06	6.48E-08
F23	1.54E-02	2.61E-06	9.73E-08	6.33E-08	4.65E-03	1.32E-07	6.03E-08	4.65E-03	1.32E-07	6.03E-08
F24	2.96E-04	3.44E-06	6.43E-08	6.19E-08	4.63E-04	2.73E-07	6.27E-08	4.36E-04	2.37E-07	6.72E-08
F25	6.33E-08	6.11E-04	2.62E-04	6.73E-08	6.42E-02	6.23E-08	6.97E-08	6.38E-02	6.72E-08	6.56E-08
F26	2.41E-03	8.27E-03	3.35E-05	6.26E-08	8.97E-04	3.95E-06	6.25E-08	8.32E-04	3.44E-06	6.52E-08
F27	6.76E-06	1.75E-03	2.73E-07	6.31E-08	1.83E-04	2.92E-07	6.53E-08	1.24E-04	2.47E-07	6.61E-08
F28	6.71E-08	1.32E-07	6.46E-07	6.28E-08	3.35E-05	1.46E-07	6.71E-08	3.47E-05	1.62E-07	6.65E-08
F29	1.24E-07	1.64E-01	3.28E-06	6.37E-08	5.54E-01	1.73E-06	6.22E-08	5.53E-01	1.37E-06	6.22E-08
F30	6.54E-08	1.32E-01	1.28E-07	6.32E-08	6.18E-08	6.91E-08	6.83E-08	6.78E-08	6.41E-08	6.33E-08

Table 7. Scalability Test Average results of all algorithms over 30 functions.

F		CSA	EHO	GOA	HHO	LSHADE	Lshade-EpSin	MFO	MVO	PSO	AO	mAO
	10	4.34E+07	6.24E+07	4.46E+07	4.65E+07	3.15E+08	2.23E+09	4.74E+08	3.64E+08	4.37E+06	5.235E+06	2.87E+05

F1 *Continue on the next page*

Table 7. Scalability Test Average results of all algorithms over 30 functions.

F		CSA	EHO	GOA	HHO	LSHADE	Lshade-EpSin	MFO	MVO	PSO	AO	mAO
	30	7.39E+09	8.31E+09	5.64E+09	5.64E+09	4.05E+10	1.36E+11	3.80E+10	2.79E+10	5.49E+08	5.45E+08	1.78E+07
	50	9.79E+11	9.42E+11	7.77E+11	6.86E+11	5.23E+11	2.19E+13	2.83E+12	4.85E+12	7.65E+10	6.62E+10	2.63E+09
F3	10	1.19E+04	6.73E+07	2.07E+04	2.32E+04	2.21E+04	3.87E+04	2.66E+04	2.45E+04	3.60E+04	3.23E+04	3.19E+04
	30	1.32E+05	8.74E+09	2.18E+05	1.80E+05	3.38E+05	2.75E+05	3.92E+05	1.75E+05	2.71E+05	1.58E+05	1.31E+05
	50	1.65E+07	6.99E+10	3.46E+07	2.32E+07	3.44E+06	3.23E+06	5.77E+07	2.43E+07	4.55E+07	3.21E+07	3.11E+07
F4	10	2.88E+02	7.88E+08	2.32E+02	3.75E+02	7.22E+02	2.42E+03	7.03E+02	4.66E+02	6.67E+02	3.11E+02	1.21E+02
	30	2.02E+03	9.03E+09	1.18E+03	1.91E+03	8.15E+03	3.92E+04	4.21E+03	5.24E+03	8.93E+02	8.36E+02	5.91E+02
	50	2.62E+05	5.29E+11	3.25E+04	3.77E+05	6.22E+05	7.13E+06	7.55E+05	7.31E+05	6.21E+05	2.44E+04	2.31E+04
F5	10	7.53E+02	5.93E+09	7.83E+02	8.92E+02	1.10E+03	1.32E+03	9.46E+02	1.06E+03	7.58E+02	8.18E+02	7.26E+02
	30	8.64E+02	9.26E+09	9.51E+02	9.45E+02	1.13E+03	1.36E+03	1.00E+03	1.08E+03	8.43E+02	8.64E+02	7.77E+02
	50	4.91E+01	2.37E+09	6.86E+01	3.90E+01	6.41E+01	4.90E+01	7.50E+01	6.10E+01	7.57E+01	4.44E+01	2.55E+01
F6	10	4.70E3+02	7.39E+07	4.36E+02	5.79E+02	4.64E+02	5.41E+02	4.51E+02	5.27E+02	4.31E+02	4.13E+02	3.15E+02
	30	6.70E+02	9.19E+09	6.76E+02	6.79E+02	6.74E+02	7.11E+02	6.61E+02	6.53E+02	6.91E+02	6.60E+02	6.15E+02
	50	8.70E+05	4.19E+11	4.88E+02	6.99E+04	8.88E+03	8.44E+05	8.65E+04	8.23E+05	8.23E+04	4.60E+05	5.23E+04
F11	10	1.59E+03	6.31E+05	7.13E+02	8.20E+02	4.35E+03	3.11E+03	1.35E+03	2.40E+02	8.17E+02	5.22E+02	2.60E+02
	30	4.52E+03	7.36E+09	4.76E+03	3.20E+03	2.53E+04	3.05E+04	1.63E+04	8.45E+03	2.10E+03	2.65E+03	1.66E+03
	50	8.57E+05	6.36E+09	5.11E+05	4.29E+05	2.94E+05	8.75E+06	7.84E+06	7.87E+05	8.92E+05	7.83E+05	6.33E+05
F12	10	6.41E+06	4.65E+06	4.34E+05	2.66E+05	5.76E+06	5.76E+7	1.84E+06	6.56E+06	5.22E+06	3.74E+06	1.21E+04
	30	1.26E+09	8.35E+09	3.63E+08	9.33E+08	6.99E+09	5.52E+10	3.89E+09	1.17E+08	9.44E+09	5.29E+07	4.74E+07
	50	1.45E+10	8.88E+10	3.77E+10	9.83E+10	6.55E+10	5.23E+11	3.21E+10	1.45E+10	9.21E+10	5.33E+09	4.23E+10

Continue on the next page

Table 7. Scalability Test Average results of all algorithms over 30 functions.

F		CSA	EHO	GOA	HHO	LSHADE	Lshade-EpSin	MFO	MVO	PSO	AO	mAO
F13	10	1.11E+05	6.23E+06	3.11E+04	3.07E+06	6.50E+06	2.66E+07	8.03E+06	2.44E+05	2.65E+06	4.99E+03	2.33E+03
	30	1.40E+07	8.47E+09	4.43E+05	3.12E+07	1.55E+09	2.05E+10	1.33E+09	3.97E+05	2.08E+09	4.84E+04	1.76E+04
	50	1.66E+09	8.74E+10	4.34E+07	3.76E+09	7.55E+10	2.77E+11	1.66E+10	8.23E+07	2.73E+10	4.22E+07	1.45E+07
F14	10	6.35E+03	9.28E+05	6.65E+03	4.22E+05	8.13E+04	2.34E+05	3.22E+04	2.67E+04	6.17E+04	8.61E+04	6.76E+04
	30	9.35E+05	9.20E+09	6.14E+05	6.98E+06	6.92E+06	2.09E+07	3.35E+06	2.16E+05	7.00E+05	3.01E+06	4.70E+05
	50	6.53E+08	9.77E+11	6.87E+08	8.98E+08	8.72E+08	2.73E+08	3.83E+08	7.75E+08	7.76E+08	3.93E+08	4.93E+07
F21	10	2.32E+03	7.13E+07	6.17E+06	2.45E+062	2.45E+02	3.82E+02	2.28E+02	8.50E+02	2.42E+03	2.23E+03	2.12E+02
	30	2.78E+03	9.28E+09	2.78E+03	2.96E+03	2.90E+03	3.17E+03	2.77E+03	2.56E+03	2.95E+03	2.66E+03	2.62E+03
	50	2.45E+07	1.23E+10	2.11E+06	6.96E+06	2.45E+06	3.55E+07	2.27E+06	7.88E+06	2.11E+07	2.76E+05	2.23E+05
F22	10	8.37E+03	6.11E+06	6.76E+03	6.88E+06	1.33E+06	1.43E+03	1.87E+03	1.56E+04	1.56E+03	1.43E+03	2.90E+02
	30	1.07E+04	7.31E+09	1.16E+04	1.21E+04	1.77E+04	1.69E+04	1.04E+04	1.56E+04	1.53E+04	1.30E+04	9.65E+03
	50	7.27E+06	9.12E+09	5.56E+06	1.45E+06	8.57E+06	8.59E+06	6.74E+07	8.23E+06	1.21E+06	5.38E+04	9.23E+07
F23	10	3.73E+02	5.11E+08	2.34E+02	3.67E+02	3.32E+02	3.55E+02	3.65E+02	3.21E+02	3.07E+02	2.76E+03	4.03E+02
	30	3.83E+03	9.44E+09	3.34E+03	4.17E+03	3.49E+03	4.19E+03	3.16E+03	3.70E+03	3.11E+03	3.15E+03	3.03E+03
	50	3.93E+04	5.44E+10	3.13E+04	4.09E+02	4.12E+04	9.12E+04	7.19E+04	3.79E+04	3.44E+04	3.62E+04	8.83E+03
F24	10	5.04E+02	7.13E+06	1.41E+02	5.24E+02	2.43E+02	3.23E+02	4.73E+02	2.22E+02	3.55E+02	9.21E+02	2.25E+02
	30	4.05E+03	8.62E+09	3.45E+03	4.39E+03	3.65E+03	4.50E+03	3.73E+03	3.18E+03	3.33E+03	3.26E+03	3.25E+03
	50	7.23E+05	5.62E+11	5.66E+05	8.33E+05	7.23E+05	6.56E+05	2.73E+05	3.23E+06	3.26E+05	3.17E+05	5.25E+05

Table 8. Mean and Standard Deviation values obtained by various Enhanced AO.

	Average				**Standard Deviation**			
N	AOOBL	AOCLS	AORS	mAO	AOOBL	AOCLS	AORS	mAO
F1	1.23E+09	4.45E+08	1.98E+08	1.78E+07	7.63E+07	7.65E+07	7.12E+07	4.23E+06
F3	1.46E+06	2.22E+06	1.22E+06	1.31E+05	1.23E+05	4.91E+05	3.02E+06	1.55E+04
F4	1.02E+04	4.23E+03	1.31E+03	5.91E+02	6.71E+03	6.92E+03	8.71E+03	6.13E+01
F5	4.67E+03	9.78E+02	4.23E+04	7.77E+02	2.91E+02	7.12E+02	2.81E+02	2.55E+01
F6	4.71E+03	6.82E+03	6.74E+03	6.15E+02	1.19E+01	3.99E+01	7.87E+00	3.37E+00
F11	4.67E+03	2.12E+04	2.23E+04	1.66E+03	2.23E+02	1.49E+03	1.49E+03	1.19E+02
F12	5.48E+07	2.27E+08	3.27E+08	4.47E+07	6.82E+07	7.66E+07	6.23E+07	3.85E+07
F13	2.52E+04	9.22E+05	3.23E+04	1.76E+04	5.73E+04	2.40E+04	2.36E+04	6.46E+03
F14	4.78E+05	4.91E+05	6.28E+06	4.70E+05	4.62E+06	8.13E+05	8.17E+05	3.40E+05
F21	2.73E+03	3.81E+03	3.22E+04	2.62E+03	3.07E+02	1.54E+02	9.37E+01	5.30E+01
F22	6.85E+04	4.23E+04	7.23E+04	9.65E+03	6.41E+03	4.78E+03	4.04E+03	9.70E+02
F23	3.19E+03	4.27E+04	7.82E+04	3.03E+03	2.45E+02	2.39E+02	8.33E+012	8.48E+01
F24	2.52E+04	7.45E+03	6.15E+03	3.25E+03	7.68E+02	6.92E+02	3.77E+02	5.48E+01

5. Engineering problems

In this section, the performance of the developed optimizer is tested using many real-world constrained problems which contain many inequalities. These problems are Welded beam design problem, Pressure vessel design problem, Tension/compression spring design problem, Speed reducer design problem, and Three-bar truss design problem. The mathematical formulas for the above problems are existed in [68, 74, 75].

5.1. *Welded beam design problem*

The first constrained problem used in this study is welded beam design (WBD) which is proposed by Coello [76]. The aim of this problem is to find the minimum welded beam cost and its design structure is shown in Figure 4. WBD has 7 constraints and 4 design variables namely: bar thickness (b), bar height (t), weld thickness (h), and attached bar part length (l). The mathematical representation of WBD can be formulated as follows:

Consider $\vec{x} = [x_1\ x_2\ x_3\ x_4] = [h\, l\, t\, b]$

Minimize $f(\vec{x}) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14.0 + x_2)$

Subject to:

$g_1(\vec{x}) = \tau(\vec{x}) - 13600 \leq 0$

$g_2(\vec{x}) = \sigma(\vec{x}) - 30000 \leq 0$

$g_3(\vec{x}) = x_1 - x_4 \leq 0$

$g_4(\vec{x}) = 0.10471\left(x_1^2\right) + 0.04811 x_3 x_4 (14 + x_2) - 5.0 \leq 0$

$g_6(\vec{x}) = \delta(\vec{x}) - 0.25 \leq 0$

$g_7(\vec{x}) = 6000 - p_c(\vec{x}) \leq 0$

where

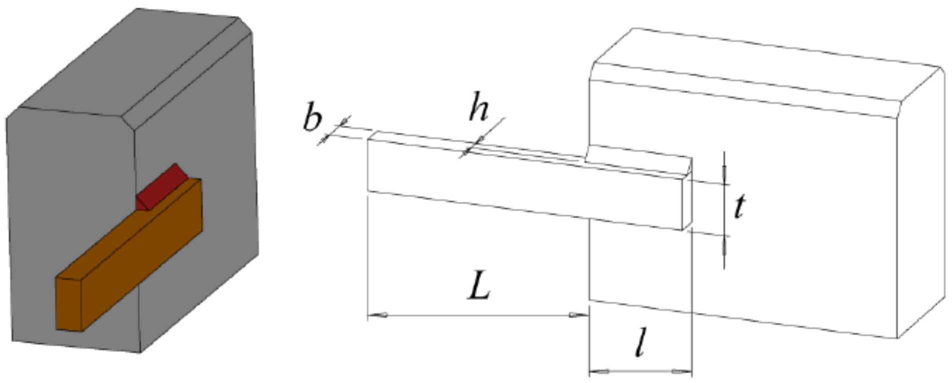


Figure 4. Welded beam design.

$\tau(\vec{x}) = \sqrt{(\tau') + (2\tau'\tau'')\frac{x_2}{2R} + (\tau'')^2}$
$\tau' = \frac{6000}{\sqrt{2}x_1x_2}$
$\tau'' = \frac{MR}{J}$
$M = 6000\left(14 + \frac{x_2}{2}\right)$
$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2}$
$j = 2\left\{x_1x_2\sqrt{2}\left[\frac{x_2^2}{12} + \left(\frac{x_1+x_3}{2}\right)^2\right]\right\}$
$\sigma(\vec{x}) = \frac{504000}{x_4x_3^2}$
$\delta(\vec{x}) = \frac{65856000}{(30\times10^6)x_4x_3^3}$
$p_c(\vec{x}) = \frac{4.013(30\times10^6)\sqrt{\frac{x_3^2x_4^6}{36}}}{196}\left(1 - \frac{x_3\sqrt{\frac{30\times10^6}{4(12\times10^6)}}}{28}\right)$
with $0.1 \le x_1, x_4 \le 2.0$ and $0.1 \le \mathrm{x}_2, \mathrm{x}_3 \le 10.0$

Results of WBD are shown in Table 9 where mAO is compared with classical AO, GSA [46], GA [77], SSA [78], MPA [79], HHO [80, 81], WOA [82], and CSA [40]. From the pre-mentioned table, it's notable that mAO has outperformed other swarm optimizers with an objective value of 1.6565 and decision values (x_1, x_2, x_3, x_4) = (0.1625, 3.4705, 9.0234, 0.2057, 1.6565).

5.2. *Pressure vessel design problem*

The 2nd constrained problem introduced in this study is one of the mixed integer optimization problems which is termed as Pressure Vessel Design (PVD) problem proposed by Kannan and Kramer [83]. PVD aims to select the lowest cost of raw materials for the cylindrical vessel as shown in Figure 5. PVD has 4 parameters namely: head thickness (T_h), shell thickness (T_s), cylindrical strength (L), and inner radius (R). To mathematically model PVD, the following formula is designed:
Minimize $f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$
Subject to:
$g_1(x) = -x_1 + 0.0193x$
$g_2(x) = -x_2 + 0.00954x_3 \le 0$

Table 9. Comparison of optimum results for Welded beam design problem.

Algorithm	***h***	***l***	***t***	***b***	**Cost**
AO	0.1631	3.3652	9.0202	0.2067	1.6566
GSA	0.1821	3.856979	10.000	0.202376	1.87995
GA	0.2489	6.1730	8.1789	0.2533	2.4300
SSA	0.2057	3.4714	9.0366	0.2057	1.72491
MPA	0.2057	3.470509	9.036624	0.205730	1.724853
SMA	0.2054	3.2589	9.0384	0.2058	1.69604
HHO	0.2134	3.5601	8.4629	0.2346	1.85614
WOA	0.3290	2.5471	6.8078	0.3789	2.358350
SO	0.2057	3.4705	9.0366	0.2057	1.72485
mAO	0.1625	3.4705	9.0234	0.2057	1.6565

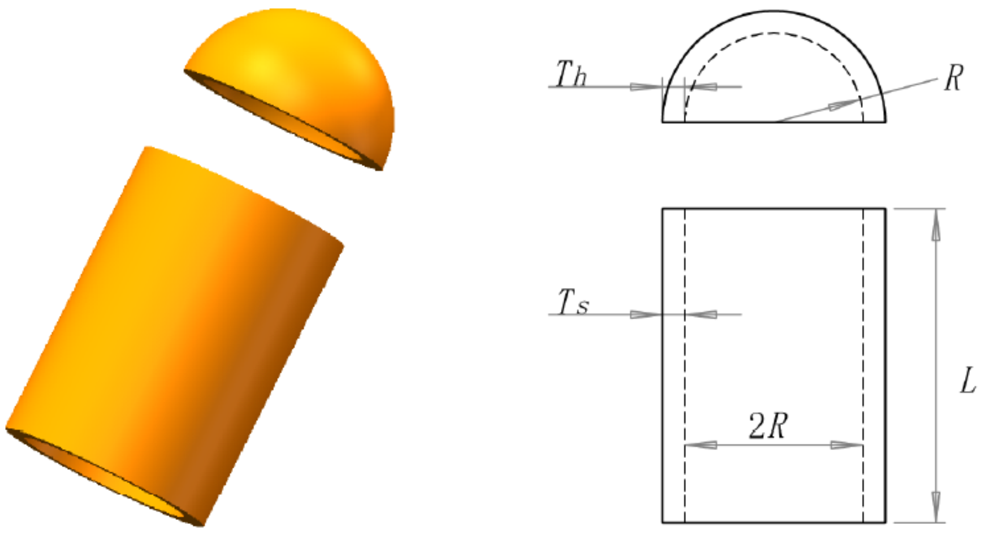


Figure 5. Pressure vessel design.

Table 10. Comparison of optimum results for pressure vessel design.

Algorithm	x_1	x_2	x_3	x_4	**Cost**
AO	1.0540	0.182806	59.6219	38.8050	5949.2258
WOA	0.812500	0.437500	42.0982699	176.638998	6059.7410
PSO-SCA	0.8125	0.4375	42.098446	176.6366	6059.71433
HS	1.125000	0.625000	58.29015	43.69268	7197.730
SMA	0.7931	0.3932	40.6711	196.2178	5994.1857
CPSO	0.8125	0.4375	42.091266	176.7465	6061.0777
GWO	0.8125	0.4345	42.0892	176.7587	6051.5639
HHO	0.81758383	0.4072927	42.09174576	176.7196352	6000.46259
GOA	0.9571	0.4749	49.9302	99.0053	6333.0873
SO	0.7819	0.3857	40.5752	196.5499	5887.5297
mAO	1.0530	0.181884	58.619	38.8080	5946.3358

$g_3(x) = -\pi x_3^2 x_4 - (4/3)\pi x_3^3 + 1,296,000 \leq 0$
$g_4(x) = x_4 - 240 \leq 0$
$0 \leq x_i \leq 100, \ i = 1,2$
$10 \leq x_i \leq 200, \ i = 3,4$
Results of PVD exists in table 10 where the suggested optimizer is compared with original AO, WOA [82], PSO-SCA [84], HS [85], SMA [86], CPSO [87], GWO [47], HHO [80], GOA [46], TEO [88], and SO [30]. From the pre-mentioned table, it can be seen that the developed optimizer ranked first with a value of 5946.3358 and decision values $(x_1, x_2, x_3, x_4) = (1.0530, 0.181884, 58.619, 38.8080, 5946.3358)$.

5.3. Tension/compression spring design problem

The 3rd constrained engineering problem discussed here is Tension/Compression Spring Design (TCSD) which was introduced by Arora [89] and its main objective is to decrease the tension spring weight by determining the optimal design variables' values that satisfy its constrained requirements. TCSD has different 3 variables namely: diameter of mean coil (D), the diameter of the wire (d), and active coils number (N). TCSD design is given in Figure 6 and its mathematical formulation is given as follows:
Consider:
$\vec{x} = [x_1 x_2 x_3] = [d\, D\, N]$
Minimize $f(\vec{x}) = (x_3 + 2)\, x_2 x_1^2$
subject to:
$g_1(\vec{x}) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0$
$g_2(\vec{x}) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} - 1 \leq 0$
$g_3(\vec{x}) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0$
$g_4(\vec{x}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0$
with $0.05 \leq x_1 \leq 2.0, 0.25 \leq x_2 \leq 1.3, and\, 2.0 \leq x_3 \leq 15.0$
Table 11 shows the results of TCSD where mAO is compared with RO [90], WOA [82], PSO [87],

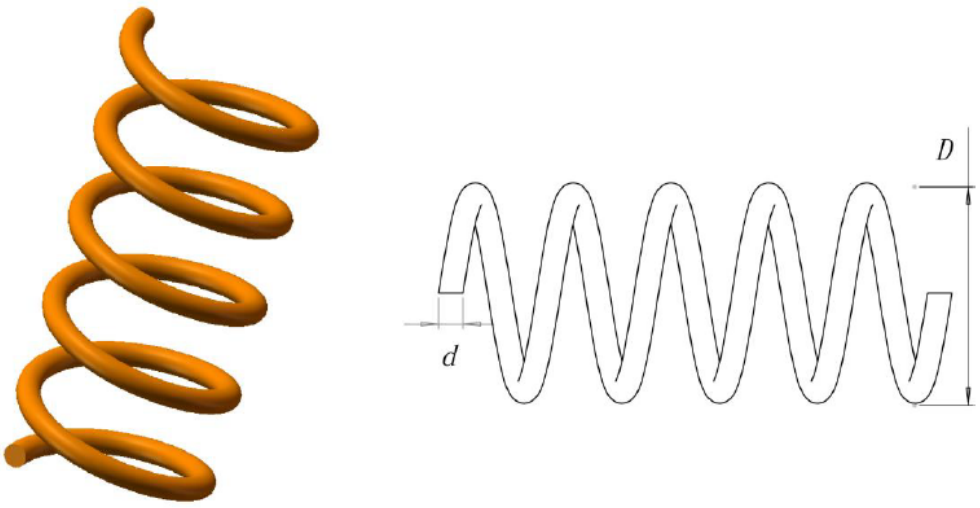


Figure 6. Tension/compression spring design.

Table 11. Comparison of optimum results for Tension/compression spring.

Algorithm	*d*	*D*	*P*	Cost
RO	0.051370	0.349096	11.76279	0.0126788
WOA	0.051207	0.345215	12.004032	0.0126763
PSO	0.051728	0.357644	11.244543	0.0126747
MVO	0.05251	0.37602	10.33513	0.012790
ES	0.051643	0.355360	11.397926	0.012698
OBSCA	0.05230	0.31728	12.54854	0.012625
GSA	0.050276	0.323680	13.525410	0.0127022
CPSO	0.051728	0.357644	11.244543	0.0126747
AO	0.0502439	0.35262	10.5425	0.011165
mAO	0.0502339	0.32282	10.5244	0.011056

MVO [64], ES [91], OBSCA [92], GSA [93], and CPSO [87].

From the previously mentioned table, we can conclude that mAO has achieved better results compared to original and other competitors. It achieves a fitness value with 0.011056 and decision values $(x_1, x_2, x_3) = (0.0502339, 0.32282, 10.5244)$.

5.4. Speed reducer design problem

The 4th engineering problem discussed in this section is the speed reducer design [94] (SRD) whose main aim is to minimize speed reducer weight with respect to curvature stress of gear teeth, shafts stress, and shafts transverse deflections. It has seven different variables and its design is shown in Figure 7. The formulation of the SRD can be descriped mathematically as follows:

$$\text{Minimize: } f(\vec{z}) = 0.7854 z_1 z_2^2 \left(3.3333 z_3^2 + 14.9334 z_3 - 43.0934\right) - 1.508 z_1 \left(z_6^2 + z_7^2\right) + 7.4777 \left(z_6^3 + z_7^3\right) + 0.7854 \left(z_4 z_6^2 + z_5 z_7^2\right)$$

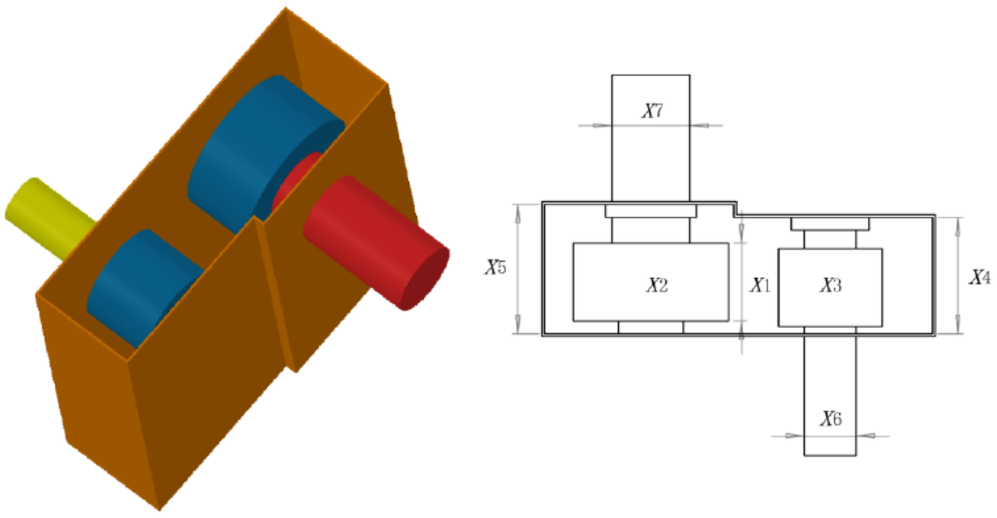


Figure 7. Speed reducer design problem.

Table 12. Comparison of optimum results for Speed Reducer problem.

Algorithm	z_1	z_2	z_3	z_4	z_5	z_6	z_7	**Cost**
Cost								
PSO	3.5001	0.7000	17.0002	7.5177	7.7832	3.3508	5.2867	3145.922
MDA	3.5	0.7	17	7.3	7.670396	3.542421	5.245814	3019.583365
GSA	3.600000	0.7	17	8.3	7.8	3.369658	5.289224	3051.120
HS	3.520124	0.7	17	8.37	7.8	3.366970	5.288719	3029.002
SCA	3.508755	0.7	17	7.3	7.8	3.461020	5.289213	3030.563
SES	3.506163	0.700831	17	7.460181	7.962143	3.362900	5.308949	3025.005127
SBSM	3.506122	0.700006	17	7.549126	7.859330	3.365576	5.289773	3008.08
hHHO-SCA	3.506119	0.7	17	7.3	7.99141	3.452569	5.286749	3029.873076
AO	3.5021	0.7000	17.0000	7.3099	7.7476	3.3641	5.2994	3007.7328
mAO	3.5012	0.7	17	7.3100	7.8873	3.0541	5.2994	3002.7328

Subject to:
$g_1(\vec{z}) = \frac{27}{z_1 z_2^2 z_3} - 1 \leq 0$
$g_2(\vec{z}) = \frac{397.5}{z_1 z_2^2 z_3} - 1 \leq 0$
$g_3(\vec{z}) = \frac{1.93 z_4^3}{z_2 z_3 z_6^4} - 1$
$g_4(\vec{z}) = \frac{1.93 z_5^3}{z_2 z_3 z_7^4} - 1 \leq 0$
$g_5(\vec{z}) = \frac{1}{110 z_6^3} \sqrt{\left(\frac{745 z_4}{z_2 z_3}\right)^2 + 16.9 \times 10^6} - 1 \leq 0$
$g_6(\vec{z}) = \frac{1}{85 z_7^3} \sqrt{\left(\frac{745 z_5}{z_2 z_3}\right)^2 + 157.5 \times 10^6} - 1 \leq 0$
$g_7(\vec{z}) = \frac{z_2 z_3}{40} - 1 \leq 0$
$g_8(\vec{z}) = \frac{5 z_2}{z_1} - 1 \leq 0$
$g_9(\vec{z}) = \frac{z_1}{12 z_2} - 1 \leq 0$
$g_{10}(\vec{z}) = \frac{1.5 z_6 + 1.9}{z_4} - 1 \leq 0$
$g_{11}(\vec{z}) = \frac{1.1 z_7 + 1.9}{z_5} - 1 \leq 0$
with$2.6 \leq z_1 \leq 3.6$
$0.7 \leq z_2 \leq 0.8$
$17 \leq z_3 \leq 28$
$7.3 \leq z_4 \leq 8.3$
$7.8 \leq z_5 \leq 8.3$
$2.9 \leq z_6 \leq 3.9$
and $5 \leq z_7 \leq 5.5$

mAO is compared with different metaheuristics optimizers including PSO [95], MDA [96], GSA [93], HS [29], SCA [97], SES [98], SBSM [99], and hHHO-SCA [100] as shown in Table 12. From this table, it's obvious that mAO outperformed other algorithms. mAO ranked first with a fitness value of 3002.7328 and decision values $(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ = (3.5012, 0.7, 17, 7.3100, 7.8873, 3.0541, 5.2994).

5.5. *Three-bar truss design problem*

The last engineering problem addressed in this manuscript is called the Three-bar truss design (TBD) problem. TBD is a fraction and nonlinear civil engineering problem introduced by Nowcki [101]. Its objective is to find the minimum values of truss weight. It has two variables and its mathematical formulation is shown below:
Minimize: $f(x) = (2\sqrt{2x_1} + x_2) * l$
Subject to: $g_1(x) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2x_1^2 + 2x_1 x_2}} P - \sigma \leq 0$
$g_2(x) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1 x_2} P - \sigma \leq 0$
$g_3(x) = \frac{1}{\sqrt{2}x_1^2 + 2x_1 x_2} P - \sigma \leq 0$
Variable Range
$0 \leq x_1, x_2 \leq 1$

mAO results are compared with CS [102], GOA [46], DEDS [103], MBA [104], PSO-DE [84],

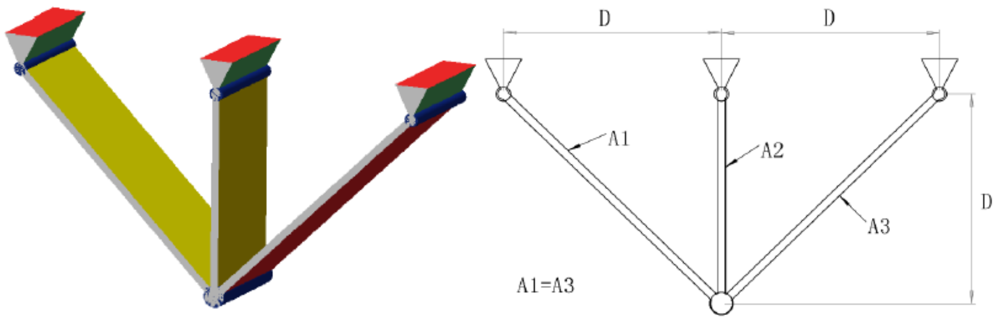


Figure 8. Three-bar truss design problem.

Table 13. Optimization results for the Three-bar truss design problem.

Algorithm	x_1	x_2	**Cost**
CS	0.78867	0.40902	263.9716
GOA	0.78889755557	0.40761957011	263.89588149
DEDS	0.78867513	0.40824828	263.89584
MBA	0.7885650	0.4085597	263.89585
PSO-DE	0.7886751	0.4082482	263.89584
AAA	0.7887354	0.408078	263.895880
CS	0.78867	0.40902	263.9716
PSO-DE	0.7886751	0.4082482	263.89584
DEDS	0.78867513	0.40824828	263.89584
AO	0.7926	0.3966	263.8684
mAO	0.7886	0.3844	231.8681

AAA [105], PSO-DE [84], DEDS [103] and AO. The results of TBD are listed in table 13 in which mAO results outperformed other competitors. mAO ranked first with a fitness value of 231.8681 and decision values (x_1, x_2) = (30.7886, 0.3844).

6. Conclusions and future work

In this study, a novel AO version is suggested called mAO to tackle various optimization issues. mAO is based on 3 different techniques: 1) Opposition-based Learning to improve optimizer exploration phase 2) Restart Strategy to remove the worse agents and replace them with totally random agents. 3) Chaotic Local Search to add more exploitation abilities to the original algorithm. mAO is tested using 29 CEC2017 functions and different five engineering optimization problems. Statistical analysis and experimental numbers show the significance of the suggested optimizer in solving various optimization issues. However, mAO like other swarm-based algorithms has a slow convergence in high-dimensional problems so, it won't be able to solve all optimization problem types.

In future, we can apply mAO to feature selection, job scheduling, combinatorial optimization problems, and stress suitability. Binary and multi-objective versions may be proposed in future.

Acknowledgments

The authors would like to thank the support of Digital Fujian Research Institute for Industrial Energy Big Data, Fujian Province University Key Lab for Industry Big Data Analysis and Application, Fujian Key Lab of Agriculture IOT Application, IOT Application Engineering Research Center of Fujian Province Colleges and Universities, Sanming City 5G Innovation Laboratory, and also the anonymous reviewers and the editor for their careful reviews and constructive suggestions to help us improve the quality of this paper. Educational research projects of young and middle-aged teachers in Fujian Province (JAT200648), Fujian Natural Science Foundation Project (2021J011128).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. Y. J. Zhang, Y. F. Wang, Y. X. Yan, J. Zhao, Z. M. Gao, Lmraoa: An improved arithmetic optimization algorithm with multi-leader and high-speed jumping based on opposition-based learning solving engineering and numerical problems, *Alexandria Eng. J.*, **61** (2022), 12367–12403. https://doi.org/10.1016/j.aej.2022.06.017
2. S. Singh, H. Singh, N. Mittal, A. G. Hussien, F. Sroubek, A feature level image fusion for night-vision context enhancement using arithmetic optimization algorithm based image segmentation, *Expert Syst. Appl.*, **209** (2022), 118272. https://doi.org/10.1016/j.eswa.2022.118272
3. A. G. Hussien, A. E. Hassanien, E. H. Houssein, M. Amin, A. T. Azar, New binary whale optimization algorithm for discrete optimization problems, *Eng. Optimiz.*, **52** (2020), 945–959. https://doi.org/10.1080/0305215X.2019.1624740
4. L. D. Giovanni, F. Pezzella, An improved genetic algorithm for the distributed and flexible job-shop scheduling problem, *Eur. J. Oper. Res.*, **200** (2010), 395–408. https://doi.org/10.1016/j.ejor.2009.01.008
5. A. G. Hussien, E. H. Houssein, A. E. Hassanien, A binary whale optimization algorithm with hyperbolic tangent fitness function for feature selection, in *IEEE 2017 Eighth international conference on intelligent computing and information systems (ICICIS)*, (2017), 166–172. https://doi.org/10.1109/INTELCIS.2017.8260031
6. A. G. Hussien, D. Oliva, E. H. Houssein, A. A. Juan, X. Yu, Binary whale optimization algorithm for dimensionality reduction, *Mathematics*, **8** (2020), 1821. https://doi.org/10.3390/math8101821
7. A. G. Hussien, M. Amin, A self-adaptive harris hawks optimization algorithm with opposition-based learning and chaotic local search strategy for global optimization and feature selection, *Int. J. Mach. Learn. Cyb.*, **13** (2022), 309–336. https://doi.org/10.1007/s13042-021-01326-4
8. Q. Liu, N. Li, H. Jia, Q. Qi, L. Abualigah, Modified remora optimization algorithm for global optimization and multilevel thresholding image segmentation, *Mathematics*, **10** (2022), 1014. https://doi.org/10.3390/math10071014

9. A. A. Ewees, L. Abualigah, D. Yousri, A. T. Sahlol, M. A. Al-qaness, S. Alshathri, et al., Modified artificial ecosystem-based optimization for multilevel thresholding image segmentation, *Mathematics*, **9** (2021), 2363. https://doi.org/10.3390/math9192363

10. M. Besnassi, N. Neggaz, A. Benyettou, Face detection based on evolutionary haar filter, *Pattern Anal. Appl.*, **23** (2020), 309–330. https://doi.org/10.1007/s10044-019-00784-5

11. E. H. Houssein, M. Amin, A. G. Hussien, A. E. Hassanien, Swarming behaviour of salps algorithm for predicting chemical compound activities, in *IEEE 2017 eighth international conference on intelligent computing and information systems (ICICIS)*, (2017), 315–320. https://doi.org/10.1109/INTELCIS.2017.8260072

12. H. Fathi, H. AlSalman, A. Gumaei, I. I. Manhrawy, A. G. Hussien, P. El-Kafrawy, An efficient cancer classification model using microarray and high-dimensional data, *Comput. Intell. Neurosci.*, **2021** (2021). https://doi.org/10.1155/2021/7231126

13. L. Abualigah, A. H. Gandomi, M. A. Elaziz, A. G. Hussien, A. M. Khasawneh, M. Alshinwan, et al., Nature-inspired optimization algorithms for text document clustering—a comprehensive analysis, *Algorithms*, **13** (2020), 345. https://doi.org/10.3390/a13120345

14. A. S. Sadiq, A. A. Dehkordi, S. Mirjalili, Q. V. Pham, Nonlinear marine predator algorithm: A cost-effective optimizer for fair power allocation in noma-vlc-b5g networks, *Expert Syst. Appl.*, **203** (2022), 117395. https://doi.org/10.1016/j.eswa.2022.117395

15. A. A. Dehkordi, A. S. Sadiq, S. Mirjalili, K. Z. Ghafoor, Nonlinear-based chaotic harris hawks optimizer: algorithm and internet of vehicles application, *Appl. Soft Comput.*, **109** (2021), 107574. https://doi.org/10.1016/j.asoc.2021.107574

16. A. S. Sadiq, A. A. Dehkordi, S. Mirjalili, J. Too, P. Pillai, Trustworthy and efficient routing algorithm for iot-fintech applications using non-linear lévy brownian generalized normal distribution optimization, *IEEE Internet Things*, 2021. https://doi.org/10.1109/JIOT.2021.3109075

17. H. Faris, S. Mirjalili, I. Aljarah, Automatic selection of hidden neurons and weights in neural networks using grey wolf optimizer based on a hybrid encoding scheme, *Int. J. Mach. Learn. Cyb.*, **10** (2019), 2901–2920. https://doi.org/10.1007/s13042-018-00913-2

18. B. Cao, J. Zhao, P. Yang, Y. Gu, K. Muhammad, J. J. Rodrigues, et al., Multiobjective 3-d topology optimization of next-generation wireless data center network, *IEEE Trans. Ind. Inf.*, **16** (2019), 3597–3605. https://doi.org/10.1109/TII.2019.2952565

19. X. Fu, P. Pace, G. Aloi, L. Yang, G. Fortino, Topology optimization against cascading failures on wireless sensor networks using a memetic algorithm, *Comput. Networks*, **177** (2020), 107327. https://doi.org/10.1016/j.comnet.2020.107327

20. L. Abualigah, A. Diabat, A comprehensive survey of the grasshopper optimization algorithm: results, variants, and applications, *Neural Comput. Appl.*, **32** (2020), 15533–15556. https://doi.org/10.1007/s00521-020-04789-8

21. H. Chen, H. Qiao, L. Xu, Q. Feng, K. Cai, A fuzzy optimization strategy for the implementation of rbf lssvr model in vis–nir analysis of pomelo maturity, *IEEE Trans. Ind. Inf.*, **15** (2019), 5971–5979. https://doi.org/10.1109/TII.2019.2933582

22. H. G. Beyer, B. Sendhoff, Robust optimization—a comprehensive survey, *Comput. Method Appl. M.*, **196** (2007), 3190–3218. https://doi.org/10.1016/J.CMA.2007.03.003

23. D. Oliva, A. A. Ewees, M. A. E. Aziz, A. E. Hassanien, M. P. Cisneros, A chaotic improved artificial bee colony for parameter estimation of photovoltaic cells, *Energies*, **10** (2017), 865. https://doi.org/10.3390/en10070865

24. J. Kennedy, R. Eberhart, Particle swarm optimization, in *IEEE Proceedings of ICNN'95-International Conference on Neural Networks*, **4** (1995), 1942–1948. https://doi.org/10.1109/ICNN.1995.488968

25. D. Karaboga, C. Ozturk, A novel clustering approach: Artificial bee colony (abc) algorithm, *Appl. Soft Comput.*, **11** (2011), 652–657. https://doi.org/10.1016/j.asoc.2009.12.025

26. R. R. Mostafa, A. G. Hussien, M. A. Khan, S. Kadry, F. A. Hashim, Enhanced coot optimization algorithm for dimensionality reduction, in *IEEE 2022 Fifth International Conference of Women in Data Science at Prince Sultan University (WiDS PSU)*, (2022), 43–48. https://10.1109/WiDS-PSU54548.2022.00020

27. J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, MIT press, 1992.

28. A. H. Gandomi, A. H. Alavi, Krill herd: A new bio-inspired optimization algorithm, *Commun. Nonlinear Sci.*, **17** (2012), 4831–4845. https://10.1016/j.cnsns.2012.05.010

29. Z. W. Geem, J. H. Kim, G. V. Loganathan, A new heuristic optimization algorithm: harmony search, *Simulation*, **76** (2001), 60–68. https://doi.org/0037-5497(2001)l:2¡60:ANHOAH¿2.0.TX;2-3

30. F. A. Hashim, A. G. Hussien, Snake optimizer: A novel meta-heuristic optimization algorithm, *Knowl.-Based Syst.*, **242** (2022), 108320. https://doi.org/10.1016/j.knosys.2022.108320

31. G. G. Wang, S. Deb, Z. Cui, Monarch butterfly optimization, *Neural Comput. Appl.*, **31** (2019), 1995–2014. https://doi.org/10.1007/s00521-015-1923-y

32. S. Li, H. Chen, M. Wang, A. A. Heidari, S. Mirjalili, Slime mould algorithm: A new method for stochastic optimization, *Future Gener. Comput. Syst.*, **111** (2020), 300–323. https://doi.org/10.1016/j.future.2020.03.055

33. G. G. Wang, Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems, *Memet. Comput.*, **10** (2018), 151–164. https://doi.org/10.1007/s12293-016-0212-3

34. Y. Yang, H. Chen, A. A. Heidari, A. H. Gandomi, Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts, *Expert Syst. Appl.*, **177** (2021), 114864. https://doi.org/10.1016/j.eswa.2021.114864

35. I. Ahmadianfar, A. A. Heidari, A. H. Gandomi, X. Chu, H. Chen, Run beyond the metaphor: An efficient optimization algorithm based on runge kutta method, *Expert Syst. Appl.*, **181** (2021), 115079. https://doi.org/10.1016/j.eswa.2021.115079

36. I. Ahmadianfar, A. A. Heidari, S. Noshadian, H. Chen, A. H. Gandomi, Info: An efficient optimization algorithm based on weighted mean of vectors, *Expert Syst. Appl.*, **195** (2022), 116516. https://doi.org/10.1016/j.eswa.2022.116516

37. A. G. Hussien, A. A. Heidari, X. Ye, G. Liang, H. Chen, Z. Pan, Boosting whale optimization with evolution strategy and gaussian random walks: an image segmentation method, *Eng. Comput.*, (2022), 1–45. https://doi.org/10.1007/s00366-021-01542-0

38. L. Abualigah, M. A. Elaziz, A. G. Hussien, B. Alsalibi, S. M. J. Jalali, A. H. Gandomi, Lightning search algorithm: A comprehensive survey, *Appl. Intell.*, **51** (2021), 2353–23760. https://doi.org/10.1007/s10489-020-01947-2

39. A. S. Assiri, A. G. Hussien, M. Amin, Ant lion optimization: variants, hybrids, and applications, *IEEE Access*, **8** (2020), 77746–77764. https://doi.org/10.1109/ACCESS.2020.2990338

40. A. G. Hussien, M. Amin, M. Wang, G. Liang, A. Alsanad, A. Gumaei, et al., Crow search algorithm: Theory, recent advances, and applications, *IEEE Access*, **8** (2020), 173548–173565. https://doi.org/10.1109/ACCESS.2020.3024108

41. A. G. Hussien, M. Amin, M. A. E. Aziz, A comprehensive review of moth-flame optimisation: variants, hybrids, and applications, *J. Exp. Theor. Artif.*, **32** (2020), 705–725. https://doi.org/10.1080/0952813X.2020.1737246

42. R. Zheng, A. G. Hussien, H. M. Jia, L. Abualigah, S. Wang, D. Wu, An improved wild horse optimizer for solving optimization problems, *Mathematics*, **10** (2022), 1311. https://doi.org/10.3390/math10081311

43. S. Wang, A. G. Hussien, H. Jia, L. Abualigah, R. Zheng, Enhanced remora optimization algorithm for solving constrained engineering optimization problems, *Mathematics*, **10** (2022), 1696. https://doi.org/10.3390/math10101696

44. L. Wang, Q. Cao, Z. Zhang, S. Mirjalili, W. Zhao, Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems, *Eng. Appl. Artif. Intell.*, **114** (2022), 105082. https://doi.org/10.1016/j.engappai.2022.105082

45. W. Zhao, Z. Zhang, S. Mirjalili, L. Wang, N. Khodadadi, S. M. Mirjalili, An effective multi-objective artificial hummingbird algorithm with dynamic elimination-based crowding distance for solving engineering design problems, *Comput. Method. Appl. M.*, **398** (2022), 115223. https://doi.org/10.1016/j.cma.2022.115223

46. S. Saremi, S. Mirjalili, A. Lewis, Grasshopper optimisation algorithm: theory and application, *Adv. Eng. Software.*, **105** (2017), 30–47. https://doi.org/10.1016/j.advengsoft.2017.01.004

47. S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Software*, **69** (2014), 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

48. E. H. Houssein, A. G. Hussien, A. E. Hassanien, S. Bhattacharyya, M. Amin, S-shaped binary whale optimization algorithm for feature selection, in *First International Symposium on Signal and Image Processing (ISSIP 2017)*, 2017. 79–87.

49. L. Abualigah, D. Yousri, M. A. Elaziz, A. A. Ewees, M. A. Al-Qaness, A. H. Gandomi, Aquila optimizer: a novel meta-heuristic optimization algorithm, *Comput. Ind. Eng.*, **157** (2021), 107250. https://doi.org/10.1016/j.cie.2021.107250

50. S. Wang, H. Jia, L. Abualigah, Q. Liu, R. Zheng, An improved hybrid aquila optimizer and harris hawks algorithm for solving industrial engineering optimization problems, *Processes*, **9** (2021), 1551. https://doi.org/10.3934/mbe.2021352

51. S. Mahajan, L. Abualigah, A. K. Pandit, M. Altalhi, Hybrid aquila optimizer with arithmetic optimization algorithm for global optimization tasks, *Soft Comput.*, **26** (2022), 4863–4881. https://doi.org/10.1007/s00500-022-06873-8

52. L. Abualigah, A. Diabat, S. Mirjalili, M. A. Elaziz, A. H. Gandomi, The arithmetic optimization algorithm, *Comput. Methods Appl. Mech. Eng.*, **376** (2021), 113609. https://doi.org/10.1016/j.cma.2020.113609

53. Y. J. Zhang, Y. X. Yan, J. Zhao, Z. M. Gao, Aoaao: The hybrid algorithm of arithmetic optimization algorithm with aquila optimizer, *IEEE Access*, **10** (2022), 10907–10933. https://doi.org/10.1109/ACCESS.2022.3144431

54. J. Zhao, Z. M. Gao, H. F. Chen, The simplified aquila optimization algorithm, *IEEE Access*, **10** (2022), 22487–22515. https://doi.org/10.1109/ACCESS.2022.3153727

55. C. Ma, H. Huang, Q. Fan, J. Wei, Y. Du, W. Gao, Grey wolf optimizer based on aquila exploration method, *Expert Syst. Appl.*, **205** (2022), 117629. https://doi.org/10.1016/j.eswa.2022.117629

56. B. Gao, Y. Shi, F. Xu, X. Xu, An improved aquila optimizer based on search control factor and mutations, *Processes*, **10** (2022), 1451. https://doi.org/10.3390/pr10081451

57. A. M. AlRassas, M. A. Al-qaness, A. A. Ewees, S. Ren, M. A. Elaziz, R. Damaševičius, et al., Optimized anfis model using aquila optimizer for oil production forecasting, *Processes*, **9** (2021), 1194. https://doi.org/10.3390/pr9071194

58. M. A. Elaziz, A. Dahou, N. A. Alsaleh, A. H. Elsheikh, A. I. Saba, M. Ahmadein, Boosting covid-19 image classification using mobilenetv3 and aquila optimizer algorithm, *Entropy*, **23** (2021), 1383. https://doi.org/10.3390/e23111383

59. A. Fatani, A. Dahou, M. A. Al-Qaness, S. Lu, M. A. Elaziz, Advanced feature extraction and selection approach using deep learning and aquila optimizer for iot intrusion detection system, *Sensors*, **22** (2021), 140. https://doi.org/10.3390/s22010140

60. G. G. Wang, S. Deb, L. D. S. Coelho, Elephant herding optimization, in *IEEE 2015 3rd International Symposium on Computational and Business Intelligence (ISCBI)*, (2015), 1–5. https://doi.org/10.1109/ISCBI.2015.8

61. R. Tanabe, A. S. Fukunaga, Improving the search performance of shade using linear population size reduction, in *2014 IEEE Congress on Evolutionary Computation (CEC)*, (2014), 1658–1665. https://doi.org/10.1109/CEC.2014.6900380

62. N. H. Awad, M. Z. Ali, P. N. Suganthan, Ensemble sinusoidal differential covariance matrix adaptation with euclidean neighborhood for solving cec2017 benchmark problems, in *2017 IEEE Congress on Evolutionary Computation (CEC)*, (2017), 372–379. https://doi.org/10.1109/CEC.2017.7969336

63. S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, *Knowl.-Based Syst.*, **89** (2015), 228–249. https://doi.org/10.1016/j.knosys.2015.07.006

64. S. Mirjalili, S. M. Mirjalili, A. Hatamlou, Multi-verse optimizer: a nature-inspired algorithm for global optimization, *Neural Comput. Appl.*, **27** (2016), 495–513. https://doi.org/10.1007/s00521-015-1870-7

65. K. Steenhof, M. N. Kochert, T. L. Mcdonald, Interactive effects of prey and weather on golden eagle reproduction, *J. Anim. Ecol.*, **66** (1997), 350–362.

66. H. R. Tizhoosh, Opposition-based learning: a new scheme for machine intelligence, in *IEEE International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, **1** (2005), 695–701. https://doi.org/10.1109/CIMCA.2005.1631345

67. A. G. Hussien, An enhanced opposition-based salp swarm algorithm for global optimization and engineering problems, *J. Amb. Intell. Hum. Comput.*, **13** (2022), 129–150. https://doi.org/10.1007/s12652-021-02892-9

68. H. Chen, Y. Xu, M. Wang, X. Zhao, A balanced whale optimization algorithm for constrained engineering design problems, *Appl. Math. Modell.*, **71** (2019), 45–59. https://doi.org/10.1016/j.apm.2019.02.004

69. Y. Yu, S. Gao, S. Cheng, Y. Wang, S. Song, F. Yuan, Cbso: A memetic brain storm optimization with chaotic local search, *Memet. Comput.*, **10** (2018), 353–367. https://doi.org/10.1007/s12293-017-0247-0

70. J. Zhao, Y. Zhang, S. Li, Y. Wang, Y. Yan, Z. Gao, A chaotic self-adaptive jaya algorithm for parameter extraction of photovoltaic models, *Math. Biosci. Eng.*, **19** (2022), 5638–5670. https://doi.org/10.3934/mbe.2022264

71. H. Zhang, Z. Wang, W. Chen, A. A. Heidari, M. Wang, X. Zhao, et al., Ensemble mutation-driven salp swarm algorithm with restart mechanism: Framework and fundamental analysis, *Expert Syst. Appl.*, **165** (2021), 113897. https://doi.org/10.1016/j.eswa.2020.113897

72. Y. Zhang, Y. Wang, S. Li, F. Yao, L. Tao, Y. Yan, et al., An enhanced adaptive comprehensive learning hybrid algorithm of rao-1 and jaya algorithm for parameter extraction of photovoltaic models, *Math. Biosci. Eng.*, **19** (2022), 5610–5637. https://doi.org/10.3934/mbe.2022263

73. Y. J. Zhang, Y. X. Yan, J. Zhao, Z. M. Gao, Cscahho: Chaotic hybridization algorithm of the sine cosine with harris hawk optimization algorithms for solving global optimization problems, *Plos One*, **17** (2022), e0263387. https://doi.org/10.1371/journal.pone.0263387

74. M. Y. Cheng, D. Prayogo, A novel fuzzy adaptive teaching–learning-based optimization (fatlbo) for solving structural optimization problems, *Eng. Comput.*, **33** (2017), 55–69. https://doi.org/10.1007/s00366-016-0456-z

75. H. Samma, J. Mohamad-Saleh, S. A. Suandi, B. Lahasan, Q-learning-based simulated annealing algorithm for constrained engineering design problems, *Neural Comput. Appl.*, **32** (2020), 5147–5161. https://doi.org/10.1007/s00521-019-04008-z

76. C. A. C. Coello, Use of a self-adaptive penalty approach for engineering optimization problems, *Comput. Ind.*, **41** (2000), 113–127. https://doi.org/10.1016/S0166-3615(99)00046-9

77. K. Deb, Optimal design of a welded beam via genetic algorithms, *AIAA J.*, **29** (1991), 2013–2015. https://doi.org/10.2514/3.10834

78. S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, S. M. Mirjalili, Salp swarm algorithm: A bio-inspired optimizer for engineering design problems, *Adv. Eng. Software*, **114** (2017), 163–191. https://doi.org/10.1016/j.advengsoft.2017.07.002

79. A. Faramarzi, M. Heidarinejad, S. Mirjalili, A. H. Gandomi, Marine predators algorithm: A nature-inspired metaheuristic, *Expert Syst. Appl.*, **152** (2020), 113377. https://doi.org/10.1016/j.eswa.2020.113377

80. A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, *Future Gener. Comput. Syst.*, **97** (2019), 849–872. https://doi.org/10.1016/j.future.2019.02.028

81. A. G. Hussien, L. Abualigah, R. A. Zitar, F. A. Hashim, M. Amin, A. Saber, et al., Recent advances in harris hawks optimization: A comparative study and applications, *Electronics*, **11** (2022), 1919. https://doi.org/10.3390/electronics11121919

82. S. Mirjalili, A. Lewis, The whale optimization algorithm, *Adv. Eng. Software*, **95** (2016), 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008

83. B. Kannan, S. N. Kramer, An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design, *J. Mech. Design*, **116** (1994), 405–411. https://doi.org/10.1115/1.2919393

84. H. Liu, Z. Cai, Y. Wang, Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization, *Appl. Soft Comput.*, **10** (2010), 629–640. https://doi.org/10.1016/j.asoc.2009.08.031

85. M. Mahdavi, M. Fesanghary, E. Damangir, An improved harmony search algorithm for solving optimization problems, *Appl. Math. Comput.*, **188** (2007), 1567–1579. https://doi.org/10.1016/j.amc.2006.11.033

86. J. Zhao, Z. M. Gao, W. Sun, The improved slime mould algorithm with levy flight, in *Journal of Physics: Conference Series*, **1617** (2020), 012033. https://doi.org/10.1088/1742-6596/1617/1/012033

87. Q. He, L. Wang, An effective co-evolutionary particle swarm optimization for constrained engineering design problems, *Eng. Appl. Artif. Intell.*, **20** (2007), 89–99. https://doi.org/10.1016/j.engappai.2006.03.003

88. A. Kaveh, A. Dadras, A novel meta-heuristic optimization algorithm: Thermal exchange optimization, *Adv. Eng. Software*, **110** (2017), 69–84. https://doi.org/10.1016/j.advengsoft.2017.03.014

89. J. S. Arora, Introduction to optimum design, Elsevier, 2004.

90. A. Kaveh, M. Khayatazad, A new meta-heuristic method: Ray optimization, *Comput. Struct.*, **112** (2012), 283–294. https://doi.org/10.1016/j.compstruc.2012.09.003

91. E. Mezura-Montes, C. A. C. Coello, An empirical study about the usefulness of evolution strategies to solve constrained optimization problems, *Int. J. Gen. Syst.*, **37** (2008), 443–473. https://doi.org/10.1080/03081070701303470

92. M. A. Elaziz, D. Oliva, S. Xiong, An improved opposition-based sine cosine algorithm for global optimization, *Expert Syst. Appl.*, **90** (2017), 484–500. https://doi.org/10.1016/j.eswa.2017.07.043

93. E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, Gsa: A gravitational search algorithm, *Inf. Sci*, **179** (2009), 2232–2248. https://doi.org/10.1016/j.ins.2009.03.004

94. E. Mezura-Montes, C. A. C. Coello, Useful infeasible solutions in engineering optimization with evolutionary algorithms, in *Mexican International Conference on Artificial Intelligence*, **3789** (2005), 652–662. https://doi.org/10.1007/11579427_66

95. S. Stephen, D. Christu, A. Dalvi, Design optimization of weight of speed reducer problem through matlab and simulation using ansys, *Int. J. Mech. Eng. Technol.*, **9** (2018), 339–349.

96. S. Lu, H. M. Kim, A regularized inexact penalty decomposition algorithm for multidisciplinary design optimization problems with complementarity constraints, *J. Mech. Design*, **132** (2010), 041005. https://doi.org/10.1115/1.4001206

97. S. Mirjalili, Sca: A sine cosine algorithm for solving optimization problems, *Knowl.-Based Syst.*, **96** (2016), 120–133. https://doi.org/10.1016/j.knosys.2015.12.022

98. E. Mezura-Montes, C. C. Coello, R. Landa-Becerra, Engineering optimization using simple evolutionary algorithm, in *Proceedings. 15th IEEE International Conference on Tools with Artificial Intelligence*, (2003), 149–156. https://doi.org/10.1109/TAI.2003.1250183

99. S. Akhtar, K. Tai, T. Ray, A socio-behavioural simulation model for engineering design optimization, *Eng. Optimiz.*, **34** (2002), 341–354. https://doi.org/10.1080/03052150212723

100. V. K. Kamboj, A. Nandi, A. Bhadoria, S. Sehgal, An intensify harris hawks optimizer for numerical and engineering optimization problems, *Appl. Soft Comput.*, **89** (2020), 106018. https://doi.org/10.1016/j.asoc.2019.106018

101. H. Nowacki, Optimization in pre-contract ship design, In *International Conference on Computer Applications in the Automation of Shipyard Operation and Ship Design*, 1973.

102. A. H. Gandomi, X. S. Yang, A. H. Alavi, Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, *Eng. Comput.*, **29** (2013), 17–35. https://doi.org/10.1007/s00366-011-0241-y

103. M. Zhang, W. Luo, X. Wang, Differential evolution with dynamic stochastic selection for constrained optimization, *Inf. Sci.*, **178** (2008), 3043–3074. https://doi.org/10.1016/j.ins.2008.02.014

104. A. Sadollah, A. Bahreininejad, H. Eskandar, M. Hamdi, Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems, *Appl. Soft Comput.*, **13** (2013), 2592–2612. https://doi.org/10.1016/j.asoc.2012.11.026

105. A. E. YILDIRIM, A. Karci, Application of three bar truss problem among engineering design optimization problems using artificial atom algorithm, in *IEEE 2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, (2018), 1–5. https://doi.org/10.1109/IDAP.2018.8620762

AIMS Press

©2022 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0)