AN ENHANCED JUMPING SPIDER OPTIMIZATION ALGORITHM

Betrand Ngwa **ATANGA**, Francis Boafo **EFFAH,** Philip Yaw **OKYERE**
Kwame Nkrumah University of Science and Technology, Kumasi, Ghana
atangabertrand@gmail.com, fbeffah74@gmail.com, okyerepy@yahoo.com

Keywords: Jumping spider optimization, bio-inspired algorithm, meta-heuristics, levy flight.

Abstract*: Jumping Spider Optimization (JSO) algorithm is a more recent meta-heuristic algorithm shown to outperform recent algorithms in the literature. This paper presents a JSO variant that modifies standard JSO in its global search and jumping on prey phases to improve further its global search capability. The proposed algorithm employs levy flight to update the position of global search agent for better performance. It also introduces a random number generator in the equation for the trajectory of the spider when jumping on prey to provide the updated agent with flexibility to attack prey. The enhanced jumping spider optimization algorithm (EJSO) is tested on 17 well-known benchmark functions and its performance compared with the standard JSO and five other well-known algorithms. The EJSO is also verified on a welded beam design problem to validate our algorithm. The results of statistical analyses conducted show the superiority of EJSO over the standard JSO and the other state-of-arts algorithms*.

1. INTRODUCTION

Several meta-heuristic algorithms have been developed to solve a variety of complex problems in a variety of domains, including data mining, engineering applications, energy, networks, medical, and other fields [1]. Meta-heuristics algorithms are flexible and straightforward because they mimic biological or physical phenomena by focusing solely on inputs and outputs [1]. Furthermore, because meta-heuristics are a type of stochastic optimization technique, they can effectively circumvent local optimum, usually encountered in real-world problems [2]. Meta-heuristic optimization algorithms outperform heuristic algorithms in various sophisticated and tricky optimization real-world problems due to the benefits of simplicity, flexibility, and the ability to avoid local optima [3]. The recent meta-heuristic algorithms include Coot Bird Algorithm (COOT) [4], Mexican Axolotl Optimization (MAO) [5], Gradient-base Optimizer (GBO) [6], Hunger Game Search (HGS) [7], and Harris Hawks Optimization (HHO) [8].

Researchers continue to develop new meta-heuristic algorithms because one algorithm is unable to obtain suitable results in all fields [9]. The jumping spider optimization algorithm (JSO), stirred by the tracking style of a jumping spider, is one of the recent additions to the existing meta-heuristic algorithms. It is shown to have the capability to solve real-world problems such as selective harmonics elimination problem and ideal tuning of parameters of a proportional integral derivative (PID) controller [10].

The optimization method presented in this paper aims at improving further the performance of the JSO. The modified version, named Enhanced Jumping Spider Optimization (EJSO) Algorithm, proves to improve the convergence rate, the efficiency in finding optimal solutions, stability and robustness of the standard JSO. The modification targets the global search and the jumping on prey phases of the JSO. The modified version introduces randomness (stochastics) into the algorithm to enable the spider agent to move in the search space efficiently and effectively to reach a good global optimization.

The remaining of the paper is organised as follows: Section 2 explains the original JSO. Section 3 presents the modified JSO. Section 4 presents the various tests done to validate the performance of the algorithm. Results and discussion are presented in section 5. Section 6 presents the conclusion.

2. ORIGINAL JSO [8]

The JSO mimics the foraging characteristics of jumping spiders. The jumping spider mathematical modelling is presented in 3 stages: searching, persecution and jumping on its prey. A model is also given for the pheromone rate of the spider.

2.1. Searching for prey

At start, the search agents are randomly generated to search for prey. In the searching for prey phase, the jumping spider undergoes a random search in the search space to find the exact position of a prey. The search is mathematically modelled in both local and global searches as depicted in *Fig. 1*.

The local search is represented by (1):

$$\vec{x}_i(k+1) = \vec{x}_{best}(k) + walk\left(\frac{1}{2} - \varepsilon\right) \tag{1}$$
$$i = 1, 2, 3, \dots, n.$$

where $\vec{x}_i(k+1)$ is the updated location of the ith search agent, $\vec{x}_{best}(k)$ is previous iteration best agent, $walk$ is a uniformly distributed pseudo-random number in the range [-2, 2] and ε is a normally distributed pseudo-random number in the range [0,1].

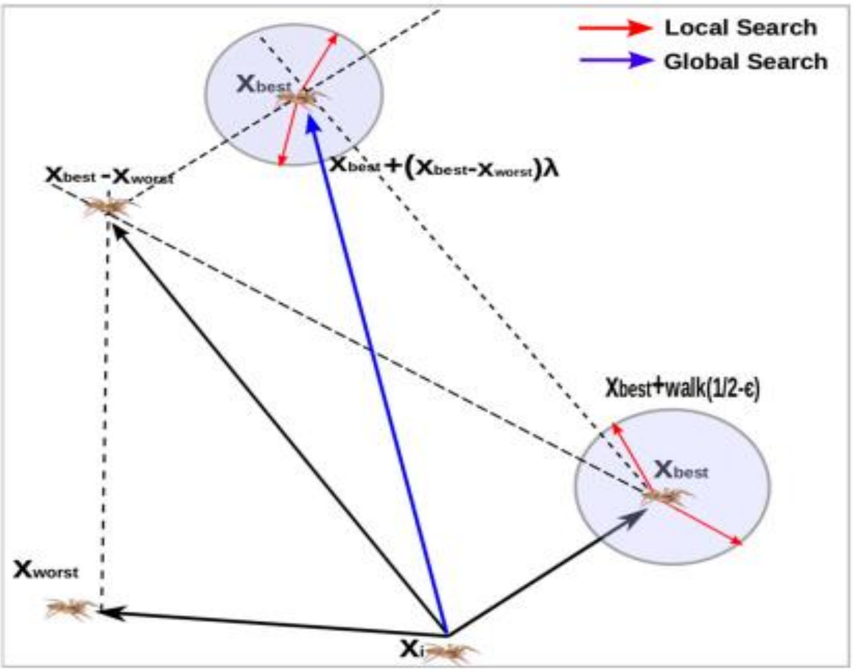


Fig. 1. Search phase

The global search is described by (2):

$$\vec{x}_i(k+1) = \vec{x}_{best}(k) + \left(\vec{x}_{best}(k) - \vec{x}_{worst}(k)\right)\lambda \quad (2)$$
$$i = 1,2,3 \ldots \ldots . . n.$$

where $\vec{x}_i(k+1)$ is updated location of ith search agent, $\vec{x}_{best}(k)$ and $\vec{x}_{worst}(k)$ indicate best and worst search agents from previous iteration, λ is a Cauchy arbitrary figure with μ=0 and θ=1.

2.2. Persecution

During hunting, the spider may find itself not within a reachable distance to capture a prey. It will creep closer until it is within a good range to jump and capture the prey. The movement is represented by uniformly accelerated rectilinear motion given by (3).

$$x_i = \frac{1}{2}at^2 + v_o t \quad (3)$$

where x_i is the position of the ith spider chasing a prey, t is time, and v_o is the starting speed. Acceleration a can be expressed as $a = \frac{v}{t}$, where $v = x - x_0$. Each iteration is taken to be time with the difference from one iteration to the next being 1. The starting speed is usually made zero and (3) redefined as in (4):

$$\vec{x}_i(k+1) = \frac{1}{2}\left(\vec{x}_i(k) - \vec{x}_r(k)\right) \tag{4}$$

where $\vec{x}_i(k+1)$ is the updated location of ith search agent, $\vec{x}_i(k)$ being the ith search agent in the previous iteration, with $\vec{x}_r(k)$ being the rth search agent arbitrarily picked from the previous iteration. The integer r lies in the interval [1, n] where n is maximum number of search agents and it must not be equal to i. The persecution is depicted as shown in *Fig. 2*.

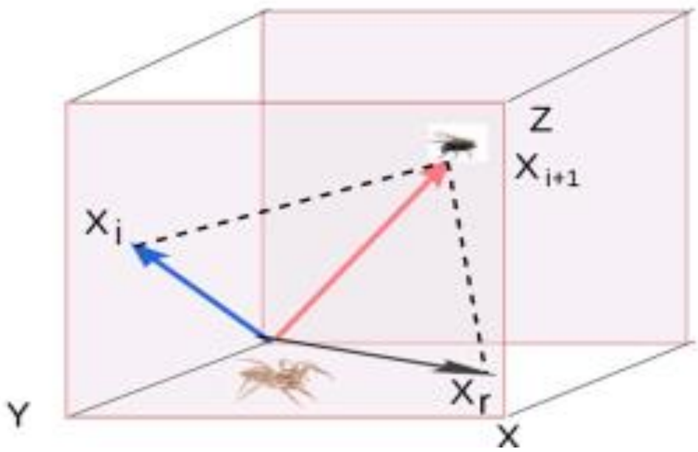


Fig. 2. Representation of persecution

2.3. Jumping on the prey

When the spider is within a jumping distance of the prey, it jumps on it. The jumping motion is considered to be a projectile motion shown in *Fig. 3*.

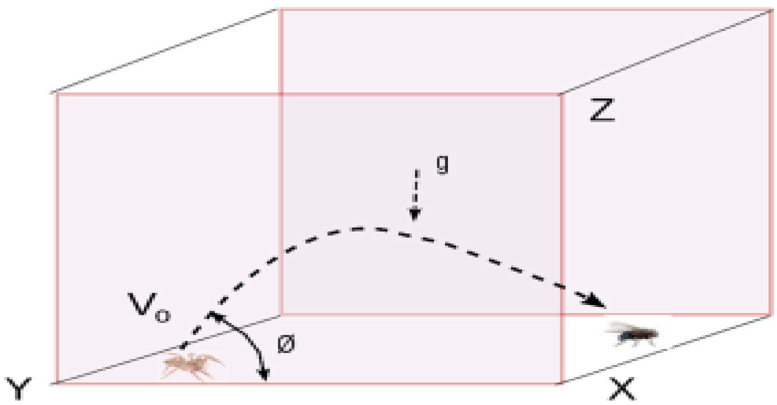


Fig. 3. Jumping on prey

The equation of projectile motion can be expressed in terms of the vertical (Y-axis) and horizontal (X-axis) displacements of the particle. The displacement along the X-axis has uniform rectilinear motion, and the Y-axis has uniformly accelerated motion. Equations (5) and (6) represent X-axis and Y-axis displacements respectively.

$$\vec{x}_i = v_o \cos(\alpha) t \vec{\imath} \tag{5}$$

$$\vec{y}_i = \left(v_0 \sin(\alpha) t - \frac{1}{2} g t^2\right) \vec{\jmath} \tag{6}$$

Eliminating the time *t* from (5) and (6), the equation of the trajectory of the projectile becomes

$$y = x \tan(\alpha) - \frac{g x^2}{2 V_0^2 cos^2(\alpha)} \tag{7}$$

The trajectory in its final form is expressed as follows:

$$\vec{x}_i(k+1) = \vec{x}_i(k) \tan(\alpha) - \frac{g \vec{x}_i^2(k)}{2 V_0^2 cos^2(\alpha)} \tag{8}$$

$$\alpha = \frac{\varphi \pi}{180}$$

where $\vec{x}_i(k+1)$ is the new location of *i*th search agent, with $\vec{x}_i(k)$ being the current location of *i*th search agent. The projection speed v_o is fixed as 100 mm/sec, g (acceleration due to gravity) = 9.80665 m/s^2 and φ in degrees is randomly generated between 0 and 1.

2.4. Pheromone Rates

Jumping spiders produce pheromones. Pheromones are olfactorily noticed by other members of the same species and they cause behavioural changes. The rate of pheromones is modelled as follows:

$$pheromone(i) = \frac{Fitness_{max} - Fitness(i)}{Fitness_{max} - Fitness_{min}} \tag{9}$$

Fitness(i) defines the present fitness value of the ith search agent, and $Fitness_{max}$ and $Fitness_{min}$ are the worst and the best fitness value in the current generation, respectively. The fitness value is normalized in the interval (0, 1) with 0 being the worst and 1 the best pheromone rate, respectively.

An unfit agent with low pheromone rate, less or equal to 0.3 is updated by a better fitted agent as follows:

$$\vec{x}_i(k) = \vec{x}_{best}(k) + \frac{1}{2}\left(\vec{x}_{r1}(k) - (-1)^{\sigma} * \vec{x}_{r2}(k)\right) \qquad (10)$$
$$r_1 \neq r_2$$

where $\vec{x}_i(k)$ is the weak jumping spider search agent with low pheromone to be updated, r_1 and r_2 are randomly generated integers in the interval [1, n], n is the maximum number of search agents, $\vec{x}_{r1}(k)$ and $\vec{x}_{r2}$ are the r_1th and r_2th search agents selected, $\vec{x}_{best}(k)$ is the best search agent found from the previous iteration and σ is a randomly generated binary number in the interval [0,1].

3. THE PROPOSED JSO VARIANT

The JSO variant is achieved by modifying the searching for prey and jumping on prey equations of the original JSO.

3.1. Modification of Searching for Prey Phase

The global search of the standard JSO is represented mathematically by (2). The second term generates new solutions around the best solution using the difference between the best solution and worst solution times the Cauchy random variable with μ= 0 and θ=1. The modified JSO replaces this term with Lévy flight which has been successfully applied for global search in many metaheuristic algorithms [9]. The new equation is given by:

$$\vec{x}_i(k+1) = \vec{x}_{best}(k) + \gamma . L_{\alpha}(S) \qquad (11)$$

where $\vec{x}_i(k+1)$ is the new location of ith search agent, $\gamma > 0$ is the step size which relates to the scales of the problem and $L_{\alpha}(S)$ provides a random walk whose random step length S is drawn from a Lévy distribution. The subscript α determines the probability of obtaining Lévy random numbers in the tail of the Lévy distribution. The choice of α significantly affects the search ability [11][12]. Its recommended value of 1.5 is used [12].

3.2. Jumping on Prey

The path followed by the spider as it jumps on prey is represented by a projectile path. In the standard JSO, the path is varied by randomly varying the projection angle φ in degrees in the interval (0, 1). For small values of α, (7) can be rewritten as

$$y = x\,\alpha - \frac{gx^2}{2V_0^2} \qquad (12)$$

From (12), the variation of φ in this narrow range produces very little effect on the first term and negligible effect on the second term. Hence not enough randomness is provided for efficient local search. The modified JSO applies another random generator to each of the two terms for more efficient exploitation. The new update equation for jumping on prey is given by:

$$\vec{x}_i(k+1) = \left(\vec{x}_i(k)\, tan(\alpha) - \frac{g\vec{x}_i^2(k)}{2V_0^2 cos^2(\alpha)}\right) * \mu \tag{13}$$

$$\alpha = \frac{\varphi\pi}{180}$$

where $\vec{x}_i(k+1)$ is the new position of *i*th search agent, $\vec{x}_i(k)$ is the current position of *i*th search agent and both φ and μ are randomly generated numbers in range (0, 1).

The solution procedure of the EJSO is similar to the standard JSO except for the modified equations for the global search and the jumping trajectory. The steps taken are given in the flowchart shown in *Fig. 4* [13].

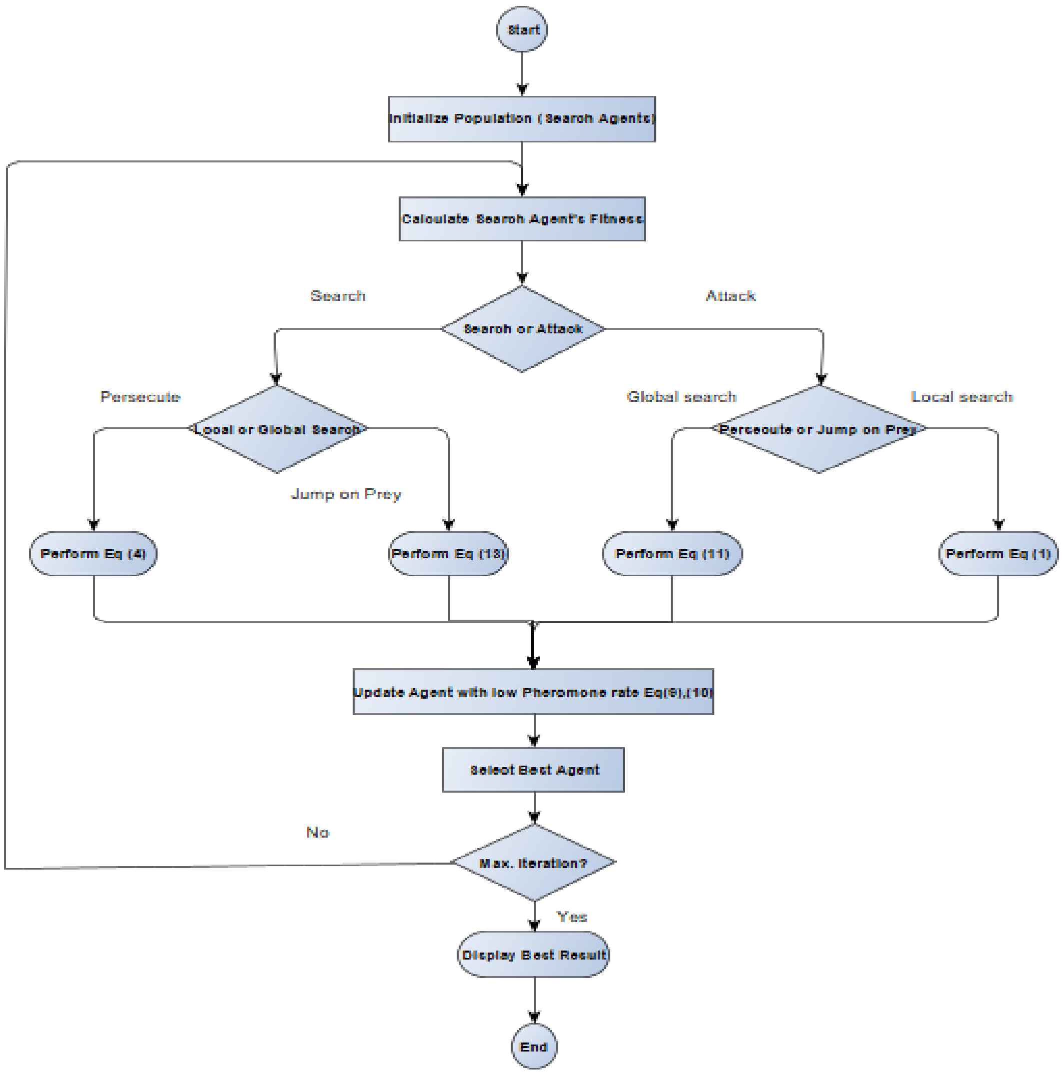


Fig. 4. Flow chart of EJOA

4. TESTING

This section presents the various computational experiments used to evaluate the performance of the proposed EJSO. The EJSO was tested on 17 standard benchmark optimization functions and welded beam design problem as an example of a real-world single objective bound constrained numerical optimization problem [8]. The results obtained were compared with the results of the standard JSO and five other recent state-of-the-art bioinspired algorithms from the literature.

4.1. Benchmark Optimization Functions

Details of the benchmark functions with varying levels of complexity are listed in Table 1. Dim indicates the dimension of the function.

Table 1. Benchmark functions

FCN	Function Name	Search Range	Dim	Optimum Value
F1	Sphere	[-100, 100]	10	0
F2	Schwefel 2.21	[-100, 100]	30	0
F3	Rotated Hyper-Ellipsoid	[65.536,65.536]	2	0
F4	Schwefel 2.22	[-100, 100]	30	0
F5	Rosenbrock	[-5, 10]	30	0
F6	Step Function	[-5.12, 5.12]	30	0
F7	Ackley	[-1,1]	30	0
F8	Beale	[-4.5,4.5]	2	0
F9	Happy cat	[-2,2]	30	0
F10	Matyas	[-10, 10]	2	0
F11	Powel	[-1,1]	50	0
F12	Salomon	[-100, 100]	50	0
F13	Colville	[-10,10]	4	0
F14	Griewank	[-600,600]	30	0
F15	Easom function	[-5,10]	30	0
F16	Sum of difference	[-1,1]	2	0
F17	Sum Square	[-10,10]	50	0

The following parameters were used for all algorithms:

Number of runs = 100

Population of search agents = 100

Maximum number of iterations = 500

Three of the algorithms require additional parameters. The algorithms and their extra parameters are given in Table 2.

Table 2. Additional parameter settings for MAO, GBO & HHO

Algorithms	**Parameters**	**Value**
	Crossover probability (*cop*)	0.5
MAO	Damage probability (*dp*)	0.5
	Regeneration probability (*rp*)	0.1
	Tournament size (*k*) Lamba value (*l*)	2, 0.5
GBO	βmin, βmax	0.2, 0.6
	pr	0.5
HHO	Harris Hawk Number	30
	E0 variable changes from -1 to 1 (Default)	

For each benchmark function, the optimum solution value, standard deviation, mean value and convergence curve were obtained for each of the seven algorithms.

The quality of the mean values obtained by each algorithm in all the 17 functions was further statistically determined using the Mean Absolute Error (MAE) given by:

$$MAE = \sum_{i=1}^{n} \frac{|O_i - P_i|}{n} \tag{14}$$

where O_i is the mean of the optimal values yielded by an algorithm for a test function Fi, P_i is the function optimal value and *n* is the number of the test functions.

4.2.Welded Beam Design Problem.

The main objective of this problem is to minimize the cost of manufacturing and to obtain best possible construction cost for the following constraint variables [13]: thickness of weld (x_1), height (x_2), length (x_3) and bar thickness (x_4). The mathematical formulation is as follows:

$\vec{x} = [x_1\ x_2\ x_3\ x_4]$

Minimize $f(\vec{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)$,

Subject to

$g_1(\vec{x}) = x_1 - x_4 \leq 0$
$g_2(\vec{x}) = \delta(\vec{x}) - \delta_{\max} \leq 0$
$g_3(\vec{x}) = P \leq P_c(\vec{x})$
$g_4(\vec{x}) = \tau_{max} \geq \tau(\vec{x})$
$g_5(\vec{x}) = \sigma(\vec{x}) - \sigma_{max} \leq 0$
Having bounds:
$0.125 \leq x_1 \leq 2$
$0.1 \leq x_2 x_3 \leq 10$
$0.1 \leq x_1, x_4 \leq 2$
Where,

$$\tau(\vec{x}) = \sqrt{(\tau')^2 + (\tau'')^2 + 2\tau'\tau''\frac{x_2}{2R}}$$

$$\tau' = \frac{P}{\sqrt{2}x_1x_2}, \tau'' = \frac{MR}{J}, M = P\left(L + \frac{x_2}{2}\right)$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2}$$

$$J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + \frac{x_1 + x_3}{2}\right]\right\}$$

$$\sigma(\vec{x}) = \frac{6PL}{x_4x_3^3}$$

$L = 14in,$ P=6000lb, E=30.10^6 psi, $\sigma_{max} = 30{,}000\ psi,$
$\tau_{max} = 13{,}600\ psi$, $\delta_{\max} = 0.25in.$

All experiments were carried out in MATLAB 2019a on a PC with Intel (R) Celeron (R) CPU N3050 @1.60GHz and a 4GB RAM memory on windows 10 OS.

5. RESULTS AND DISCUSSION

5.1. Results for Benchmark Functions

Comparison test results for the benchmark functions are presented in Tables 3 - 6. The convergence curves of the seven algorithms are compared for each test function in *Fig. 5*.

Table 3. Optimal Values.

FCN	EJSO	JSO	COOT	GBO	HGS	HHO	MAO
F1	**0.00x10^{00}**	3.96x10^{-152}	1.57x10^{-64}	8.17x10^{-199}	**0.00x10^{00}**	2.93x10^{-141}	3.87x10^{-01}
F2	1.78x10^{-181}	2.62x10^{-93}	1.39x10^{-29}	1.73x10^{-100}	**0.00x10^{00}**	3.44x10^{-72}	7.06x10^{-02}
F3	**0.00x10^{00}**	2.62x10^{-151}	9.78x10^{-30}	2.13x10^{-190}	**0.00x10^{00}**	2.77x10^{-118}	1.33x10^{+00}
F4	8.87x10^{-179}	5.57x10^{-94}	5.46x10^{-29}	2.68x10^{-99}	**0.00x10^{00}**	2.05x10^{-65}	6.27x10^{-02}

FCN	EJSO	JSO	COOT	GBO	HGS	HHO	MAO
F5	**$0.00x10^{00}$**	$1.97x10^{-31}$	$1.76x10^{-14}$	**$0.00x10^{00}$**	$6.44x10^{-22}$	**$0.00x10^{00}$**	$4.34x10^{-01}$
F6	**$0.00x10^{00}$**	**$0.00x10^{00}$**	$5.91x10^{-27}$	**$0.00x10^{00}$**	**$0.00x10^{00}$**	$1.18x10^{-10}$	$3.93x10^{-01}$
F7	**$-8.88x10^{-16}$**	$-8.88x10^{-16}$	$6.22x10^{-15}$	**$-8.88x10^{-16}$**	**$-8.88x10^{-16}$**	**$-8.88x10^{-16}$**	$1.35x10^{+00}$
F8	**$0.00x10^{00}$**	**$0.00x10^{00}$**	$1.12x10^{-20}$	$0.00x10^{+00}$	**$0.00x10^{00}$**	**$0.00x10^{00}$**	$9.73x10^{-03}$
F9	$2.85x10^{-02}$	$2.50x10^{-01}$	$1.02x10^{-02}$	$6.58x10^{-03}$	$1.61x10^{-03}$	**$2.05x10^{-04}$**	$3.05x10^{-01}$
F10	**$0.00x10^{00}$**	$2.05x10^{-150}$	$8.26x10^{-67}$	$1.96x10^{-176}$	**$0.00x10^{00}$**	$8.88x10^{-155}$	$1.10x10^{-05}$
F11	**$0.00x10^{00}$**	**$0.00x10^{00}$**	**$0.00x10^{00}$**	**$0.00x10^{00}$**	**$0.00x10^{00}$**	**$0.00x10^{00}$**	$1.53x10^{-207}$
F12	$9.99x10^{-02}$	$1.20x10^{-90}$	$9.56x10^{-11}$	$6.77x10^{-59}$	**$0.00x10^{00}$**	$3.00x10^{-54}$	$1.50x10^{+01}$
F13	**$0.00x10^{00}$**	$1.09x10^{-11}$	$5.73x10^{-09}$	**$0.00x10^{00}$**	$3.62x10^{-06}$	$1.69x10^{-07}$	$1.49x10^{+01}$
F14	**$0.00x10^{00}$**	**$0.00x10^{00}$**	**$0.00x10^{00}$**	**$0.00x10^{00}$**	**$0.00x10^{00}$**	**$0.00x10^{00}$**	$1.83x10^{+00}$
F15	$6.46x10^{-191}$	$2.19x10^{-144}$	$6.73x10^{-48}$	$3.24x10^{-156}$	**$0.00x10^{00}$**	$8.36x10^{-121}$	$1.08x10^{+00}$
F16	**$0.00x10^{00}$**	$3.86x10^{-164}$	$1.09x10^{-43}$	$2.95x10^{-226}$	$6.07x10^{-260}$	$1.20x10^{-140}$	$4.20x10^{-05}$
F17	$6.46x10^{-191}$	$2.74x10^{-148}$	$2.49x10^{-45}$	$3.48x10^{-151}$	**$0.00x10^{00}$**	$1.28x10^{-118}$	$3.98x10^{+02}$

Table 4. Mean Values.

FNC NO.	EJSO	JSO	COOT	GBO	HGS	HHO	MAO
F1	**$8.42x10^{-06}$**	$1.07x10^{-02}$	$1.58x10^{-01}$	$1.15x10^{-02}$	$2.34x10^{-02}$	$9.48x10^{-03}$	$1.79x10^{+02}$
F2	**$2.81x10^{-04}$**	$7.32x10^{-04}$	$4.62x10^{-03}$	$1.22x10^{-03}$	$1.71x10^{-03}$	$5.80x10^{-04}$	$1.39x10^{+00}$
F3	**$1.62x10^{-05}$**	$1.42x10^{-02}$	$9.96x10^{-02}$	$1.87x10^{-02}$	$2.82x10^{-02}$	$1.39x10^{-02}$	$2.15x10^{+02}$
F4	**$1.78x10^{-04}$**	$7.28x10^{-03}$	$1.57x10^{-02}$	$9.88x10^{-03}$	$1.41x10^{-02}$	$4.90x10^{-03}$	$5.25x10^{+00}$
F5	**$2.52x10^{-03}$**	$1.30x10^{-01}$	$2.94x10^{-01}$	$1.00x10^{+00}$	$1.74x10^{-01}$	$9.29x10^{-02}$	$5.34x10^{+04}$
F6	**$8.95x10^{-06}$**	$6.47x10^{-03}$	$1.34x10^{-01}$	$1.10x10^{-02}$	$1.37x10^{-02}$	$6.81x10^{-03}$	$1.76x10^{+02}$
F7	**$7.74x10^{-04}$**	$2.03x10^{-02}$	$4.66x10^{-02}$	$1.49x10^{-02}$	$2.57x10^{-02}$	$1.09x10^{-02}$	$5.45x10^{+00}$
F8	$2.93x10^{-02}$	$1.75x10^{-03}$	**$1.50x10^{-03}$**	$5.35x10^{-04}$	$1.78x10^{-03}$	$1.75x10^{-03}$	$9.00x10^{+01}$
F9	$4.34x10^{-02}$	$2.91x10^{-01}$	$1.47x10^{-01}$	$4.46x10^{-02}$	$2.87x10^{-02}$	**$1.28x10^{-02}$**	$5.67x10^{-01}$
F10	**$3.39x10^{-07}$**	$3.44x10^{-05}$	$3.35x10^{-04}$	$2.09x10^{-05}$	$3.68x10^{-05}$	$1.78x10^{-05}$	$1.26x10^{-01}$
F11	$3.03x10^{-154}$	$8.39x10^{-135}$	$2.49x10^{-160}$	**$3.80x10^{-202}$**	$3.37x10^{-135}$	$3.36x10^{-135}$	$2.24x10^{-18}$
F12	$1.04x10^{-01}$	**$7.28x10^{-02}$**	$4.94x10^{-01}$	$1.08x10^{-01}$	$1.40x10^{-01}$	$1.25x10^{-01}$	$2.32x10^{+01}$
F13	**$4.24x10^{-01}$**	$1.77x10^{+00}$	$1.58x10^{+00}$	$6.02x10^{-01}$	$2.13x10^{+00}$	$1.57x10^{+00}$	$2.73x10^{+03}$
F14	**$9.99x10^{-05}$**	$3.66x10^{-02}$	$7.18x10^{-02}$	$3.01x10^{-02}$	$5.15x10^{-02}$	$4.44x10^{-02}$	$1.05x10^{+01}$
F15	**$3.46x10^{-03}$**	$4.78x10^{-02}$	$1.07x10^{-01}$	$2.10x10^{-02}$	$1.58x10^{-01}$	$6.44x10^{-02}$	$1.45x10^{+01}$
F16	$5.20x10^{-06}$	$1.46x10^{-07}$	$2.51x10^{-06}$	$1.15x10^{-07}$	$2.40x10\text{-}^{07}$	**$4.24x10^{-08}$**	$1.44x10^{-02}$
F17	**$3.46x10^{-03}$**	$1.77x10^{+01}$	$3.95x10^{+01}$	$7.80x10^{+00}$	$5.79x10^{+01}$	$2.38x10^{+01}$	$5.34x10^{+03}$

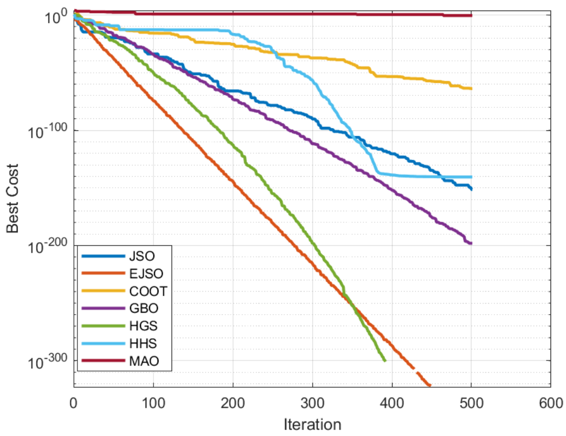
Table 5. Standard Deviation

FNC N.	EJSO	JSO	COOT	GBO	HGS	HHO	MAO
F1	**$1.32x10^{-04}$**	$2.13x10^{-01}$	$2.00x10^{+00}$	$2.12x10^{-01}$	$3.02x10^{-01}$	$2.11x10^{-01}$	$8.28x10^{+02}$
F2	**$3.98x10^{-03}$**	$1.22x10^{-02}$	$5.36x10^{-02}$	$1.39x10^{-02}$	$1.76x10^{-02}$	$1.14x10^{-02}$	$4.75x10^{+00}$
F3	**$2.96x10^{-04}$**	$2.80x10^{-01}$	$1.15x10^{+00}$	$2.99x10^{-01}$	$4.00x10^{-01}$	$2.80x10^{-01}$	$1.25x10^{+03}$
F4	**$2.31x10^{-03}$**	$1.13x10^{-01}$	$2.59x10^{-01}$	$1.35x10^{-01}$	$1.45x10^{-01}$	$9.62x10^{-02}$	$1.09x10^{+01}$
F5	**$5.33x10^{-02}$**	$2.17x10^{+00}$	$4.19x10^{+00}$	$1.13x10^{+00}$	$2.51x10^{+00}$	$1.97x10^{+00}$	$7.02x10^{+05}$
F6	**$1.16x10^{-04}$**	$1.25x10^{-01}$	$2.13x10^{+00}$	$1.45x10^{-01}$	$1.77x10^{-01}$	$1.24x10^{-01}$	$8.32x10^{+02}$
F7	**$1.14x10^{-02}$**	$2.54x10^{-01}$	$4.13x10^{-01}$	$1.71x10^{-01}$	$2.64x10^{-01}$	$1.60x10^{-01}$	$4.16x10^{+00}$
F8	$6.52x10^{-01}$	$3.76x10^{-02}$	$1.87x10^{-02}$	**$1.19x10^{-02}$**	$3.77x10^{-02}$	$3.77x10^{-02}$	$1.30x10^{+03}$

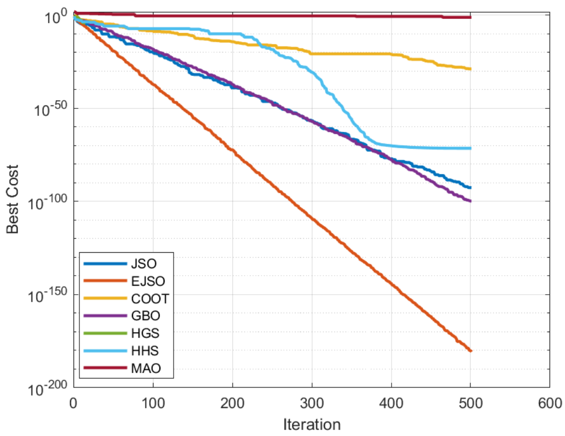
FNC N.	EJSO	JSO	COOT	GBO	HGS	HHO	MAO
F9	1.95x10^{-01}	6.63x10^{-02}	2.42x10^{-01}	8.22x10^{-02}	9.16x10^{-02}	**5.96x10^{-02}**	8.41x10^{-01}
F10	**4.05x10^{-06}**	5.44x10^{-04}	4.64x10^{-03}	4.04x10^{-04}	5.63x10^{-04}	3.99x10^{-04}	4.79x10^{-01}
F11	4.80x10^{-153}	1.19x10^{-133}	**0.00x10^{+00}**	**0.00x10^{+00}**	7.52x10^{-134}	7.50x10^{-134}	5.07x10^{-17}
F12	**2.73x10^{-02}**	1.49x10^{+00}	2.36x10^{+00}	5.50x10^{-01}	1.66x10^{+00}	1.68x10^{+00}	7.48x10^{+00}
F13	**3.46x10^{+00}**	3.32x10^{+01}	2.01x10^{+01}	7.55x10^{+00}	3.40x10^{+01}	3.31x10^{+01}	1.63x10^{+04}
F14	**2.05x10^{-03}**	7.47x10^{-01}	2.17x10^{-01}	1.81x10^{-01}	7.65x10^{-01}	7.61x10^{-01}	1.69x10^{+01}
F15	**6.46x10^{-02}**	1.07x10^{+00}	1.24x10^{+00}	3.95x10^{-01}	1.78x10^{+00}	1.12x10^{+00}	1.68x10^{+01}
F16	7.92x10^{-05}	1.67x10^{-06}	4.68x10^{-05}	1.38x10^{-06}	2.13x10^{-06}	**9.23x10^{-07}**	7.32x10^{-02}
F17	**6.46x10^{-02}**	3.95x10^{+02}	4.60x10^{+02}	1.18x10^{+02}	6.55x10^{+02}	4.13x10^{+02}	6.22x10^{+03}

Table 6. Ranking of algorithms using MAE values

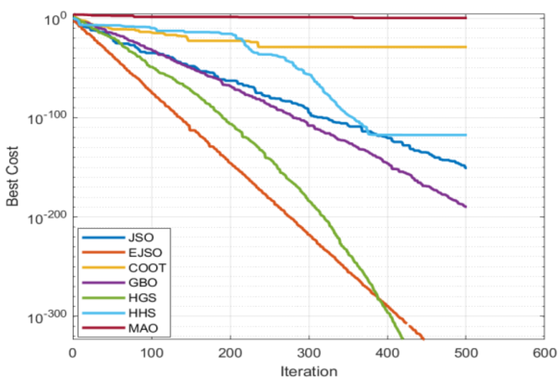
Algorithms	MAE	Rank
EJSO	3.57x10^{-02}	1
GBO	0.515906	2
JSO	1.187415	3
HHO	1.515363	4
COOT	2.508066854	5
HGS	3.482911	6
MAO	3655	7



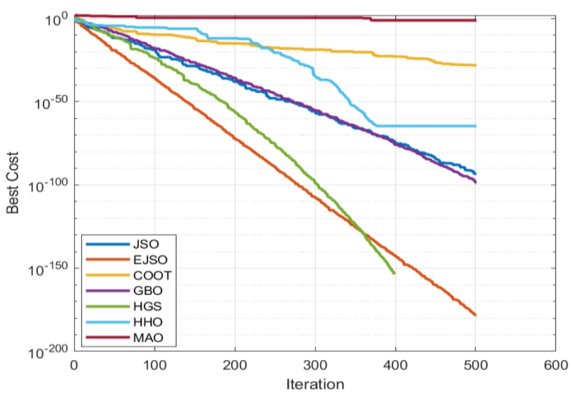
F1



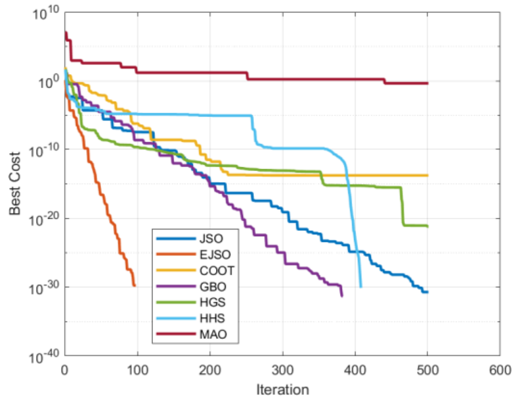
F2



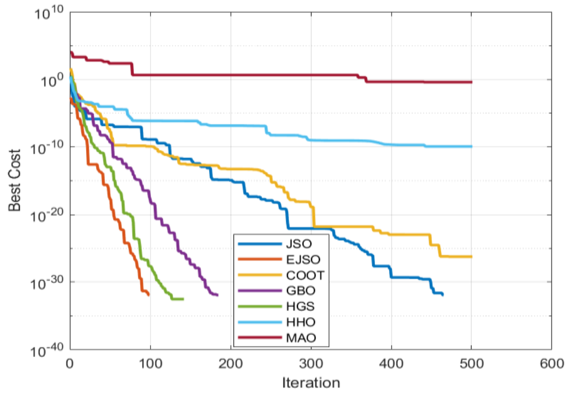
F3



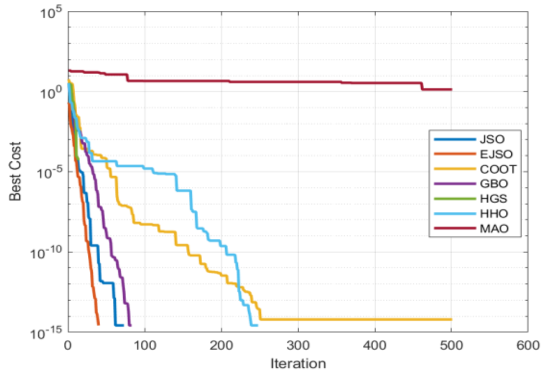
F4



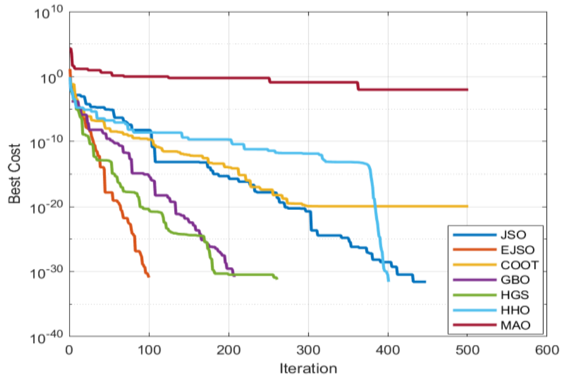
F5



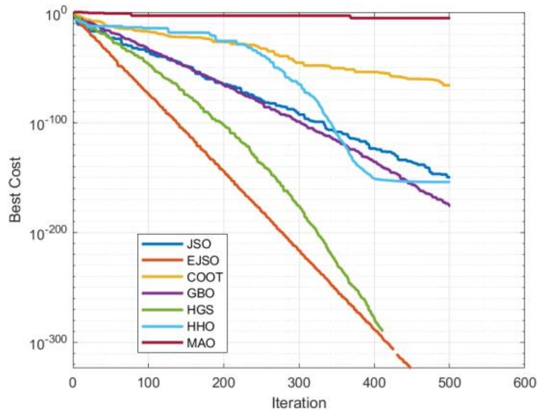
F6



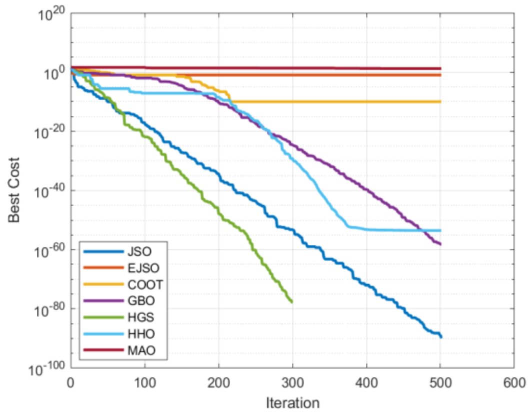
F7



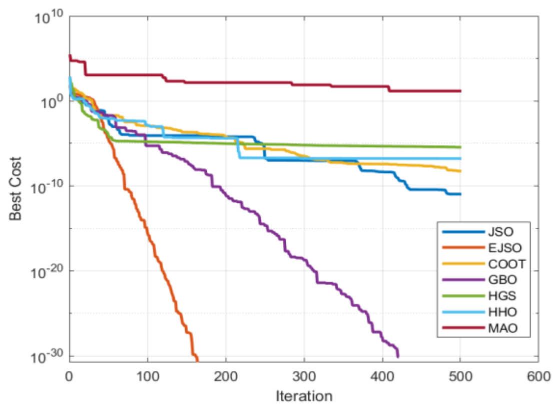
F8



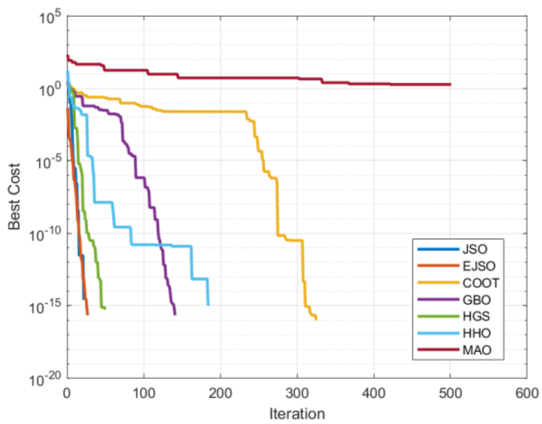
F9



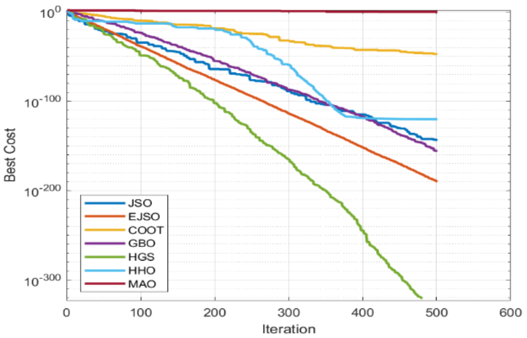
F10



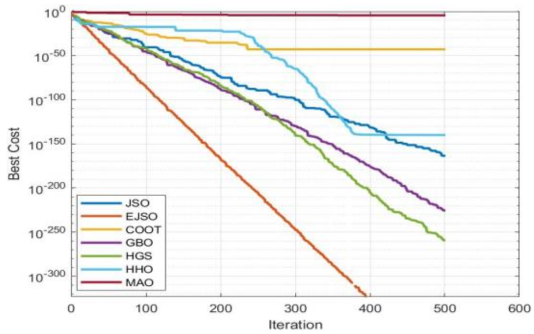
F11



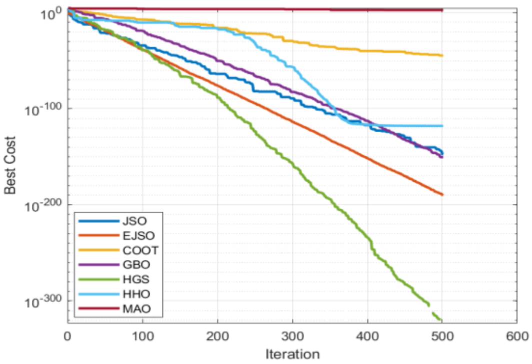
F12



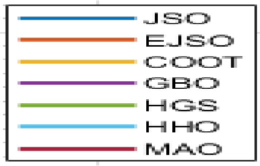
F13



F14



F15



Legend

Fig. 5. Convergence Curves.

The results in Table 3 show that the EJSO and the standard JOA were equally efficient in finding five optimal or near optimal solutions (F6, F7, F8, F11, F14) and EJSO was more efficient in finding eleven optimal or near optimal solutions (F1- F5, F9, F10, F13, F15 – F17). EJSO was less efficient only in F12. Compared to all other algorithms, it produced superior or equivalent results in eleven functions. It ranked second to only HGS which yielded superior or equivalent results in fourteen functions.

The results in Tables 4 and 5 reveal that the average optimal or near optimal solutions and standard deviations yielded by the EJSO were better in fourteen functions compared to those of the standard JSO. Compared to all other algorithms, it produced the best average optimal or near optimal solutions in twelve of the benchmark functions and the least standard deviation in thirteen of them. Compared to the competitive HGS algorithm, the EJSO yielded better mean and standard deviations in fourteen functions. The results of the statistical analyses thus show the superiority of the EJSO over all the other algorithms.

In the performance evaluation based on MAE, the EJSO ranked first as shown in Table 6. This again confirms its superiority in terms of efficiency.

From the convergence curves in *Figure 6*, EJSO is seen to converge better and faster in nine functions, namely F2, F5-F9, F11, F12, and F14, and second better in functions F1, F3, F4, F13, F15. However, its convergence in F10 was very poor. It is also noticed that the HGO presented a competitive convergence against the EJSO. Overall, the EJSO converged better than the other algorithms.

5.2. Results for Welded Beam Design

Table 7 presents results obtained from solving the welded beam problem with EJSO and the six other algorithms. From the table, EJSO outperforms all the other six algorithms. However, the result produced by the standard JSO was very competitive.

Table 8. Best cost for welded Beam

Algorithm	Optimal Values for constraint Variables				Optimal Cost
	x_1	x_2	x_3	x_4	
EJSO	0.205729	3.470418	9.036613	0.205729	1.672485
JSO	0.168168	4.514591	9.036623	0.205729	1.677014
COOT	0.198085	3.484521	9.173125	0.399772	1.879285
GBO	0.205729	3.470488	9.036623	0.205729	1.724852
HGS	0.205721	3.470666	9.036627	0.205729	1.724863
HHO	0.177143	4.309538	9.036930	0.205728	1.787066
MAO	1.036525	4.075623	6.191996	1.164902	8.041845

6. CONCLUSION

This paper has presented an enhanced Jumping Spider algorithm. The modification was done on the Global search and the Jumping on prey phases of the standard algorithm. The algorithm was tested on 17 well-known benchmark functions and a real-world optimization problem and its performance compared with the standard JSO and five other well-known algorithms in the literature. The results showed an improved performance of the algorithm in terms of convergence rate, optimal solution values, stability and robustness over the standard JSO. It also provided superior statistical results when compared to the five other state-of-the-art algorithms (COOT, GBO, HGS, HHO and MAO) from the literature.

REFERENCES

[1] S. Bandaru and K. Deb, "Metaheuristic techniques," *Decis. Sci. Theory Pract.*, pp. 693–749, 2016, doi: 10.1201/9781315183176.

[2] K. Hussain, M. N. Mohd Salleh, S. Cheng, and Y. Shi, "Metaheuristic research: a comprehensive survey," *Artif. Intell. Rev.*, vol. 52, no. 4, pp. 2191–2233, 2019, doi: 10.1007/s10462-017-9605-z.

[3] L. Abualigah and A. Diabat, "A comprehensive survey of the Grasshopper optimization

algorithm: results, variants, and applications," *Neural Comput. Appl.*, vol. 32, no. 19, pp. 15533–15556, 2020, doi: 10.1007/s00521-020-04789-8.

[4] I. Naruei and F. Keynia, "A new optimization method based on COOT bird natural life model," *Expert Syst. Appl.*, vol. 183, no. February 2020, p. 115352, 2021, doi: 10.1016/j.eswa.2021.115352.

[5] Y. Villuendas-Rey, J. L. Velázquez-Rodríguez, M. D. Alanis-Tamez, M. A. Moreno-Ibarra, and C. Yáñez-Márquez, "Mexican axolotl optimization: A novel bioinspired heuristic," *Mathematics*, vol. 9, no. 7, pp. 1–20, 2021, doi: 10.3390/math9070781.

[6] I. Ahmadianfar, O. Bozorg-Haddad, and X. Chu, "Gradient-based optimizer: A new metaheuristic optimization algorithm," *Inf. Sci. (Ny).*, vol. 540, pp. 131–159, 2020, doi: 10.1016/j.ins.2020.06.037.

[7] Y. Yang, H. Chen, A. A. Heidari, and A. H. Gandomi, "Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts," *Expert Syst. Appl.*, vol. 177, no. March, p. 114864, 2021, doi: 10.1016/j.eswa.2021.114864.

[8] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Futur. Gener. Comput. Syst.*, vol. 97, pp. 849–872, 2019, doi: 10.1016/j.future.2019.02.028.

[9] S. Mirjalili, J. Song Dong, A. Lewis, and A. S. Sadiq, *Particle swarm optimization: Theory, literature review, and application in airfoil design*, vol. 811. Springer International Publishing, 2020.

[10] H. Peraza-Vázquez, A. F. Peña-Delgado, G. Echavarría-Castillo, A. B. Morales-Cepeda, J. Velasco-Álvarez, and F. Ruiz-Perez, "A Bio-Inspired Method for Engineering Design Optimization Inspired by Dingoes Hunting Strategies," *Math. Probl. Eng.*, vol. 2021, 2021, doi: 10.1155/2021/9107547.

[11] M. Jamil and H. J. Zepernick, "Lévy Flights and Global Optimization," *Swarm Intell. Bio-Inspired Comput.*, pp. 49–72, 2013, doi: 10.1016/B978-0-12-405163-8.00003-X.

[12] X. Yang, S. Deb, and A. C. B. Behaviour, "Cuckoo Search via L ´ evy Flights," *Ieee*, pp. 210–214, 2009.

[13] H. Peraza-Vázquez, A. Peña-Delgado, P. Ranjan, C. Barde, A. Choubey, and A. B. Morales-Cepeda, "A bio-inspired method for mathematical optimization inspired by arachnida salticidade," *Mathematics*, vol. 10, no. 1, 2022, doi: 10.3390/math10010102.