Applied Soft Computing Journal 75 (2019) 775–789

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

Impact of Lèvy flight on modern meta-heuristic optimizers

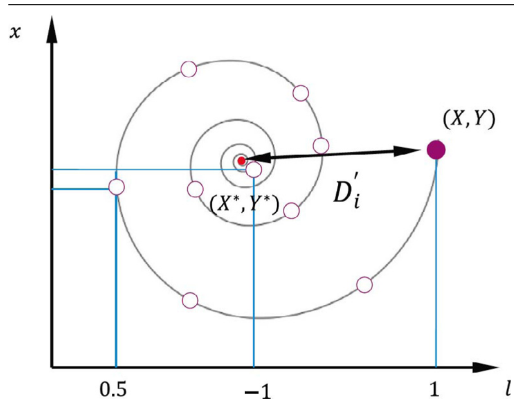
E. Emary [a,b], Hossam M. Zawbaa [c,d,*], Marwa Sharawi [b]

[a] Faculty of Computers and Information, Cairo University, Egypt
[b] Faculty of Computer Studies, Arab Open University, Cairo, Egypt
[c] Faculty of Computers and Information, Beni-Suef University, Egypt
[d] Faculty of Mathematics and Computer Science, Babes-Bolyai University, Romania

HIGHLIGHTS

- Novel modifications for the SCA and WOA are proposed.
- Random walk based on levy flight to enhance the convergence speed and global optima.
- The proposed variants were benchmarked using a set of 29 test functions,
- The evaluation is performed using a set of assessment indicators.
- Results prove the capability of our variants to outperform the original optimizers.

GRAPHICAL ABSTRACT



ARTICLE INFO

Article history:
Received 8 September 2016
Received in revised form 15 November 2018
Accepted 18 November 2018
Available online 7 December 2018

Keywords:
Whale optimization algorithm
Sine-cosine algorithm
Lèvy flight
Lèvy whale optimization algorithm
Lèvy sine-cosine algorithm

ABSTRACT

In this paper, a variant based on Lèvy flight was proposed to enhance the performance of two recently proposed optimizers. The first optimizer used in the study is Sine-Cosine Algorithm (SCA) while the second is Whale Optimization Algorithm (WOA). Both optimizers are composed of two phases of random walks in each optimization iteration and both have stagnation and premature convergence problems. Lèvy flight is used to replace the walk based on cosine function in the SCA and the spiral motion in the WOA as well. The Lèvy-based search guarantees a fraction of solutions to be generated apart from the current best solution and hence tolerates for optimizer stagnation, premature convergence, and allows for local optima avoidance. A smooth control of the scale of the Lèvy random walk is also proposed to ensure a smooth adaptation of exploration to exploitation switching. The proposed variants, as well as the original algorithms, were benchmarked using a set of unimodal, multimodal, fixed-dimension multimodal and composite benchmark functions. The evaluation is performed using a set of assessment indicators and results prove the capability of the proposed variants to outperform the original optimizers.
© 2018 Elsevier B.V. All rights reserved.

1. Introduction

Meta-heuristic optimization algorithms are recently on track of addressing some of the engineering optimization problems. It strongly proved its efficiency in optimizing combinatorial problem with larger scales as well as non-linear optimization problems that are exponentially increasing in their complexity proportional to their size. In addition, these algorithms have simple modeling and deployment of concepts with great ability to bypass local optima.

Nature-inspired meta-heuristic algorithms are a set of adaptable and flexible algorithms that are inspired by natural or physical phenomena through a population-based framework with satisfactory capabilities to handle high dimensional optimization problems. However, scientists and researchers have classified these algorithms to different inspiration categories, the swarm-based techniques of these nature-inspired optimization algorithms have been and still the most popular to reach global optimization.

Evolutionary computational (EC) techniques were applied to explore adaptively a given search space for an optimal subset of features by using a population of search agents that communicate in a social way to reach an optimal solution in a minimum number

* Corresponding author.
E-mail address: hossam.zawbaa@gmail.com (H.M. Zawbaa).

https://doi.org/10.1016/j.asoc.2018.11.033
1568-4946/© 2018 Elsevier B.V. All rights reserved.

of evaluations. In what follows, we describe shortly a set of EC methods: Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Bee Colony (ABC), Artificial Fish Swarm (AFS), Cuckoo Search (CS), Artificial Fish Swarm (AFS), and Moth Flame Optimization (MFO).

GA is extended based on the natural method of development through reproduction [1]. Solutions of PSO are sets of particles, each of which is defined by its position, fitness value, and the moving direction [2]. ABC is an optimization technique based on the foraging habits of honeybees [3]. AFS simulates the intelligence of fish movements while it is searching for food or preys [4]. CS is inspired by the generation procedure of cuckoos [5]. MFO is inspired by the moth's navigation in nature, called transverse orientation [6].

In the last few years, many researchers have studied birds and insects behavior that adopts Lèvy flights [7]. The fruit flies search their landscape applying series of straight flying paths with a 90-degree turn, reaching to a Lèvy flight interrupted scale-free search pattern. Lèvy flight method outlines an optimal random search pattern and is often observed in nature [8]. Lèvy flight is a random walk, which is defined by the sequence of quick jumps determined from a probability density function that has a power law distribution. In general, Lèvy flight is a random walk whose step lengths are drawn from the Lèvy flight distribution [9]. Therefore, Lèvy flight function has been employed to the optimization problems and the initial results prove its promising search capability [10].

CS is an EC technique that based on the Lèvy flight method. In CS optimization, producing a new solution X_i applies Lèvy flight as in Eq. (1).

$$X_i^{(t+1)} = X_i^{(t)} + \vartheta \oplus Levy(\beta), \tag{1}$$

where ϑ denotes the step size associated with the problem scale that is 1 in most cases, and the product $\oplus$ indicates entry-wise multiplications. Numerous studies have confirmed that the flying operation of many animals and insects has demonstrated the common features of Lèvy flights [11]. Lèvy flights apply a random walk while their random moves are drawn from the Lèvy distribution — more details can be found in Section 3.

The sequential jumps/steps of the cuckoo are substantially from a random walk process. It obeys the power-law step-length distribution with a thick tail.

Lèvy flights were adopted in some modern optimizers to enhance their performance. An example of such adoption is the adoption of Lèvy flights to enhance the performance of sine cosine algorithm (SCA) proposed in [12], whale optimization algorithm (WOA) in [13]. A new Lèvy flight version of antlion optimization (ALO) algorithm is introduced and used for feature selection problems in [14].

The aggregate purpose of this paper is to recommend Lèvy versions of sine cosine algorithm (LSCA) and whale optimization algorithm (LWOA) for solving 29 standard benchmark optimization test functions.

The rest of the paper is organized as follows: Section 2 shows the preliminaries and background of the sine-cosine algorithm (SCA) and whale optimization algorithm (WOA). Section 3 introduces the proposed Lèvy versions of SCA and WOA optimization algorithms. Experimental results with discussions are given in Section 4. Finally, the conclusions and future work are presented in Section 5.

2. Preliminaries

2.1. Sine-Cosine Algorithm (SCA)

SCA is a modern stochastic optimization that is based on sine and cosine methods [15]. SCA preserves the population of search agents that is represented by d-dimensions decision variable. Moreover, SCA keeps trace of the best solution's position until a given time — current best- $Leader_j^t$ accomplished by all search agents (j agents) in the population at each t iteration. The mathematical model applied in SCA is based on the sine and cosine functions as in Eqs. (2) and (3) [16].

```
begin SCA Algorithm
    Initialize a population of solutions X_i (i = 1, 2, ..., n) randomly.
    while t < T do for each iteration
        Calculate the fitness of each solution X_i.
        Update the current best solution Leader.
        Update r_1 using the equation (3).
        Randomly generate new values for r_2, r_3 and s.
        foreach search agent X_i do
            Update X_i using the equation (2).
            Check for boundary constraint according to the
            equation (4).
        end
    end
    Return Leader best solution achieved so far.
end
```

Algorithm 1: Sine cosine algorithm

$$x_{ij}^{t+1} = \begin{cases} x_{ij}^t + r_1 * sin(r_2) * |r_3 Leader_j^t - x_{ij}^t| \text{ if } r < 0.5 \\ x_{ij}^t + r_1 * cos(r_2) * |r_3 Leader_j^t - x_{ij}^t| \text{ if } r \geq 0.5, \end{cases} \tag{2}$$

$$r_1 = a - t\frac{a}{T}, \tag{3}$$

where t shows the current iteration number, T is the total number of iterations, r_1 indicates the exploration and exploitation control parameter that decreases linearly from a constant value a to 0 by each iteration according to Eq. (3), r is a random number in the range [0, 1], r_2 is another random number in the range [0, 2 π], and r_3 is a third random number in the range [0, 2] all following uniform distribution.

SCA has a circular search pattern where the best solution is in center of the circle and all other search agents are positioned around it [15] with fair opportunity for each agent to update its own position to be either between the agent's current position and *Leader*-current best- or far behind the agent's current position- X_i. The circular search area is partitioned into sub-regions that describe the possible exploration and exploitation parts of X_i. Considering the solutions always update their positions nearby the best-*Leader*- solution achieved so far, there is a bias toward the best areas of the search spaces during the optimization process. The value of r_1 controls the movement of X_i. If $r_1 > 1$ then X_i moves toward P (exploitation step), otherwise it move far away from P (exploration step) according to Eq. (3). r_2 is a random value that is used to control how far X_i moves along its direction according to r_1. r_3 assigns random weights to the best solution P which stochastically emphasizes when $r_3 > 1$ or deemphasizes when $r_3 < 1$, the desalination in determining the distance. s is a random parameter that switches between the sine and cosine parts of Eq. (2). The SCA is outlined in the algorithm 1.

Space boundary constraints are applied on all the search agents at each iteration and on all dimensions according to Eq. (4).

$$X_{i,d}^t = \begin{cases} L_d & \text{if } X_{i,d}^t < L_d \\ H_d & \text{if } X_{i,d}^t > H_d \\ X_{i,d}^t & \text{Otherwise,} \end{cases} \tag{4}$$

where L_d, H_d are the low and high boundary constraint for dimension d in order and X_i, d^t is the position of search agent number i at time t in dimension d.

2.2. Whale Optimization Algorithm (WOA)

WOA is a naturally inspired optimization method from the humpback whales' natural behavior [17]. As stated factual information, whales are naturally survived by practicing hunting behaviors. However, the power and efficiency of this algorithm over the other different methods lies in the ability of whales to work on a random or best agent in the search space in the phase of prey chasing. Moreover, whales use spirals to produce bubble-net attaching simulation mechanisms which lead to the best global optimization [18].

Encircling prey, bubble-net attacking method and search for prey are the three main behaviors of humpback whales. Modeling of these behaviors is formatted into three different phases respectively; namely; Encircling phase, exploitation phase, and exploration phase [17].

In the encircling phase, whales are trying to determine the best search agent (prey) to encircle it. WOA starts by considering that the current solution is the best one which achieved so far and assumes that it is the exact target prey or it is relatively close to the optimum solution. The other agents are all consequently updating their positions toward the determined best search agent. This can be expressed mathematically as follows:

$$\overrightarrow{D} = |\overrightarrow{C}.Leader^t - X^t|, \tag{5}$$

$$X^{t+1} = Leader^t - \overrightarrow{A}.\overrightarrow{D}, \tag{6}$$

where t indicates the current iteration, $\overrightarrow{A}$ and $\overrightarrow{C}$ are coefficient vectors. The $Leader^t$ is the position vector of the best solution achieved so far and X is the position vector. The $Leader^t$ should be updated in each iteration if there is a better solution. The vectors $\overrightarrow{A}$ and $\overrightarrow{C}$ are calculated as follows:

$$\overrightarrow{A} = 2\overrightarrow{a}.\overrightarrow{r} - \overrightarrow{a}, \tag{7}$$

$$\overrightarrow{C} = 2.\overrightarrow{r}, \tag{8}$$

where $\overrightarrow{a}$ is linearly decreased from 2 to 0 over the course of iterations and r is random vector in [0, 1].

In the exploitation phase, whales depend on the creation of bubbles along the circle. These circles are a '9' shaped path to hunt its prey. They use a spiral maneuver to dive deep down and then start to create the bubble in a spiral shape around the prey and swim up toward the surface. The exploitation is performed by two different approaches as follows:

- *Shrinking encircling mechanism*: in this step, the value of $\overrightarrow{a}$ in Eq. (7) is decreased and consequently the fluctuation range of $\overrightarrow{A}$ is decreased by $\overrightarrow{a}$ as well. This implies that $\overrightarrow{a}$ is placed as a random value in $[-\overrightarrow{a}, \overrightarrow{a}]$. a is decreased from 2 to 0 over the course of iterations. Setting random values for $\overrightarrow{A}$ in $[-1, 1]$, the new position of a search agent can be defined anywhere in between the original position of the agent and the position of the current best agent.
- *Spiral updating position*: in this step, the distance between the position of the whale and its prey is computed, and then a spiral equation is created between the position of whale and the position of the prey to mimic the helix-shaped movement of humpback whales as follows:

$$X^{t+1} = \overrightarrow{D'}.e^{bl}.\cos(2\Pi l) + X^t \tag{9}$$

$$\overrightarrow{D'} = |Leader^t - X^t| \tag{10}$$

Eq. (11) indicates the distance of the ith whale to the prey (best solution achieved so far), b is a constant for determining the shape of the logarithmic spiral, and l is a random number in $[-1, 1]$. The whale movement toward its prey concurrently applies the shrinking circling in a spiral-shaped path. Therefore, a 50% assumption of the possibility to switch between the two modes is applied to update the whale's next position as follows:

$$X^{t+1} = \begin{cases} Leader^t - \overrightarrow{A}.\overrightarrow{D} & \text{if } p < 0.5 \\ \overrightarrow{D'}.e^{bl}.\cos(2\Pi l) + Leader^t & \text{if } p \geq 0.5, \end{cases} \tag{11}$$

where p is a random number $\in$ [0, 1] following uniform distribution.

```
begin WOA Algorithm
    Initialize the whales population X_i (i = 1, 2, 3, ..., n) Compute
    the fitness of each whale.
    Set Leader as the current best search agent.
    while t < T do for each iteration
        foreach search agent X_i do
            Update a, A, C, l and p.
            if (p < 0.5) then
                if (|A| < 1) then
                    Update the position of the current search agent
                    by the equation (6).
                else
                    Select random search agent X_rand.
                    Update the position of the current search agent
                    by the equation (13).
                end
            else
                Update the position of the current search agent by
                the equation (9).
            end
            Check for boundary constraint according to equation
            (4).
        end
    end
    Return Leader best solution achieved so far.
end
```

Algorithm 2: Whale optimization algorithm

In the exploration phase, whales are reached to global optimization by searching for their prey according to their positions to each other randomly. Therefore, in order to force search agent to move far away from a reference whale, the random value of $\overrightarrow{A}$ is selected randomly out of the interval $[-1, 1]$. This means that $\overrightarrow{A}$ has to be either greater than 1 or less than the -1. Moreover, the updated position of a search agent in this phase is done according to a randomly chosen search agent which allows the WOA to perform a global search. The modeling of this exploration mechanism is mathematically expressed as follows:

$$\overrightarrow{D} = |\overrightarrow{C}.X_{rand} - X|, \tag{12}$$

$$X^{t+1} = X_{rand} - \overrightarrow{A}.\overrightarrow{D}, \tag{13}$$

where $\overrightarrow{X_{rand}}$ is a random position vector (a random whale) chosen from the current population. The pseudo code of WOA phases is presented in the Algorithm (2).

2.3. Lèvy Flight

Many types of research have studied the birds and insects flying behavior that described the typical characteristics of Lèvy flights [7]. Therefore, Lèvy flight distribution has been used in the optimization problems, and the preliminary results indicate its

Table 1
Test benchmark function examples description of CEC-2005.

The cost Func.	Dims	Variation	F_{min}	*Type*
$F_1(x) = \sum_{i=1}^{n} x_i^2$	30	$[-100, 100]$	0	Unimodal
$F_8(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$	30	$[-500, 500]$	-418.9829×5	Multimodal
$F_{14}(x) = (\frac{1}{500} \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6})^{-1}$	2	$[-65, 65]$	1	Fixed-dimension
$F_{24} = (CF1) = f_1, f_2, f_3, \ldots, f_{10} = SphereFunction$ $[\sigma_1, \sigma_2, \sigma_3, \ldots \sigma_{10}] = [1, 1, 1, \ldots, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \ldots \lambda_{10}] = [5/100, 5/100, 5/100, \ldots, 5/100]$	30	$[-5, 5]$	0	Composite

Table 2
Parameter settings for the experiments.

Parameter	Description	Value(s)
n	Number of search agents	30
T	Number of iterations	500
N_{runs}	Number of runs for each optimizer	30
β	parameter controlling the shape of Lèvy distribution	{0.5, 1, 1.5, 2}
Pa	discovery rate for CS	0.25 [5]
ω	Velocity inertia factor for PSO	Decrementing linearly from 0.9 to 0.01
c_b	individual-best acceleration factor for PSO	Decrementing linearly from 2.5 to 0.5
c_g	Global-best acceleration factor for PSO	Incremental linearly from 0.1 to 2.5

promising ability [10]. Lèvy distribution is the total of N random variables with a Fourier transform of the form (14) and the inverse Fourier transform distribution as in Eq. (15) [9]:

$$F_N(k) = \exp(-N|k|^{\beta}), \tag{14}$$

$$L(s) = \frac{1}{\pi}\int_0^{\infty} \cos(\tau s)e^{-\alpha\tau^{\beta}}d\tau, (0 < \beta \leq 2), \tag{15}$$

where $L(s)$ demonstrates Lèvy distribution with an index β, there are two special types: (a) if $\beta = 1$, the integral turns into the *Cauchy* distribution, (b) if $\beta = 2$, the integral turns into the *Gaussian* distribution.

The Eq. (15) can be expressed as an asymptotic series as in Eq. (16). The simple version of Lèvy distribution is determined in Eq. (17):

$$L(s) = |s|^{-1-\beta}, \tag{16}$$

$$L(s, \gamma, \mu) = \begin{cases} \sqrt{\frac{\gamma}{2\pi}} \exp(-\frac{\gamma}{2(s-\mu)})\frac{1}{(s-\mu)^3/2}, & 0 < \mu < s < \infty \\ 0 \; otherwise \end{cases} \tag{17}$$

where μ represents the minimum step size, and γ demonstrates the scale parameter (in our experiments, we used γ = 18). As $s \rightarrow \infty$ we have the simple form of Lèvy as in Eq. (18):

$$L(s, \gamma, \mu) = \sqrt{\frac{\gamma}{2\pi}}\frac{1}{(s-\mu)^3/2}, \tag{18}$$

3. The Proposed Lèvy flights-based variants

Randomization plays an important role in both *exploration* and *exploitation*, or *diversification* and *intensification* and the essence of

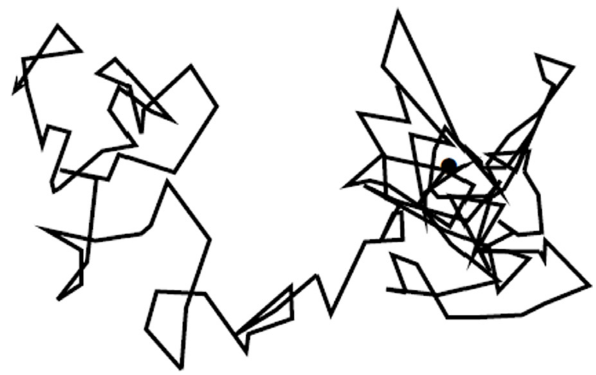


Fig. 1. Brownian motion with a gaussian step-size distribution and the path of 50 steps starting at the origin; after [19].

such randomization is the random walk [19]. A *random walk* is a random process that consists of taking a series of consecutive random steps. The random walk can be written as a sum of consecutive random steps as in Eq. (19).

$$S_N = \sum_{i=1}^{N} X_i, \tag{19}$$

where X_i is a random step drawn from a random distribution.

If the step length obeys the Gaussian distribution, the random walk becomes the Brownian motion; as shown in Fig. 1 [19].

Far-field randomization should generate a fraction of solutions and be far enough from the current best solution. Such far solutions help a given optimizer to escape from local optima and avoid stagnation [20]. A variant of Brownian motion that uses Lèvy is expected to reach that end. Lèvy distribution is a heavy-tailed distribution that has infinite variance.

According to [5], position updating following Lèvy distribution can be formulated as:

$$X_i^{t+1} = X_i^t + s.\frac{u}{|v|^{\frac{1}{\beta}}}.(X_i^t - Leader_i^t), \tag{20}$$

where X_i^t is the position of search agent in dimension i at time t, s is scale parameter suitably selected based on problem scale, β is a constant u and v are random numbers drawn from normal distribution that is:

$$u \sim \aleph(0, \sigma_u^2), v \sim \aleph(0, \sigma_v^2), \tag{21}$$

where σ_v=1 and σ_u is calculated as:

$$\sigma_u = (\frac{\Gamma(1+\beta)sin(\pi\beta/2)}{\Gamma[\frac{1+\beta}{2}]\beta 2^{\frac{\beta-1}{2}}})^{\frac{1}{\beta}}, \tag{22}$$

Table 3
Comparison of mean Performance of Lèvy variants against the original optimizers at rational values for β parameter.

Func.	$\beta = 1$				$\beta = 1.5$			
	LWOA	WOA	LSCA	SCA	LWOA	WOA	LSCA	SCA
F01	**0.0**	**0.0**	55 290	**12.7**	**0.0**	**0.0**	19 962	**12.7**
F02	**0.0**	**0.0**	**0.010**	**0.010**	**0.0**	**0.0**	**0.010**	**0.010**
F03	**24.4e+3**	52.3e+3	78 560	**9629**	**14.6e+3**	52.3e+3	61 069	**9629**
F04	**1.894**	19.849	85.595	**19.925**	**0.871**	19.849	83.844	**19.925**
F05	**25.123**	28.103	65.3e+6	**0.0e+6**	**26.867**	28.103	43.9e+5	**0.3e+5**
F06	**0.005**	0.386	52 788	**16.8**	**0.030**	0.386	24 337	**16.8**
F07	**0.001**	**0.001**	**0.003**	0.035	**0.0**	0.001	**0.003**	0.035
F08	−9244.0	**−10 378.9**	**−4546.1**	−3799.9	−9801.1	**−10 378.9**	**−4581.9**	−3799.9
F09	**0.0**	**0.0**	109.2	**40.1**	**0.0**	**0.0**	76.579	**40.109**
F10	**0.0**	**0.0**	**9.260**	10.869	**0.0**	**0.0**	**8.945**	10.869
F11	**0.025**	0.028	599.1	**0.9**	**0.015**	0.028	575.5	**0.9**
F12	**0.001**	0.029	40.1e+7	**0.0e+7**	0.062	**0.029**	58.4e+6	**0.1e+6**
F13	**0.109**	0.489	86.5e+7	**0.0e+7**	**0.169**	0.489	17.3e+7	**0.0e+7**
F14	**0.998**	3.015	**1.049**	1.594	**0.998**	3.015	**0.998**	1.594
F15	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**
F16	**−1.032**	**−1.032**	**−1.032**	**−1.032**	**−1.032**	**−1.032**	**−1.032**	**−1.032**
F17	**0.398**	**0.398**	**0.399**	0.400	**0.398**	**0.398**	**0.399**	0.400
F18	**3.0**	**3.0**	**3.0**	**3.0**	**3.0**	**3.0**	**3.0**	**3.0**
F19	**−3.852**	−3.850	**−3.854**	−3.853	**−3.853**	−3.850	**−3.854**	−3.853
F20	**−3.307**	−3.271	**−3.068**	−3.057	**−3.296**	−3.271	**−3.062**	−3.057
F21	−6.717	**−7.793**	**−2.826**	−2.669	−7.085	**−7.793**	**−3.321**	−2.669
F22	−7.341	**−7.984**	**−3.516**	−2.709	**−8.272**	−7.984	**−4.183**	−2.709
F23	−6.466	**−7.393**	**−4.639**	−4.258	−7.159	**−7.393**	**−4.568**	−4.258
F24	**106.3**	140.9	**133.4**	137.7	**75.6**	140.9	144.4	**137.7**
F25	**140.7**	182.3	**114.6**	115.8	199.8	**182.3**	**111.5**	115.8
F26	**321.1**	411.7	**377.1**	424.6	**385.7**	411.7	**370.4**	424.6
F27	**496.2**	545.0	**427.3**	452.0	564.0	**545.0**	**439.1**	452.0
F28	**90.7**	209.3	**129.5**	130.6	**131.1**	209.3	**126.1**	130.6
F29	**716.8**	754.9	**550.7**	578.6	**752.2**	754.9	**566.4**	578.6

Table 4
Comparison of mean Performance of Lèvy variants against the original optimizers at outlier values for β parameter.

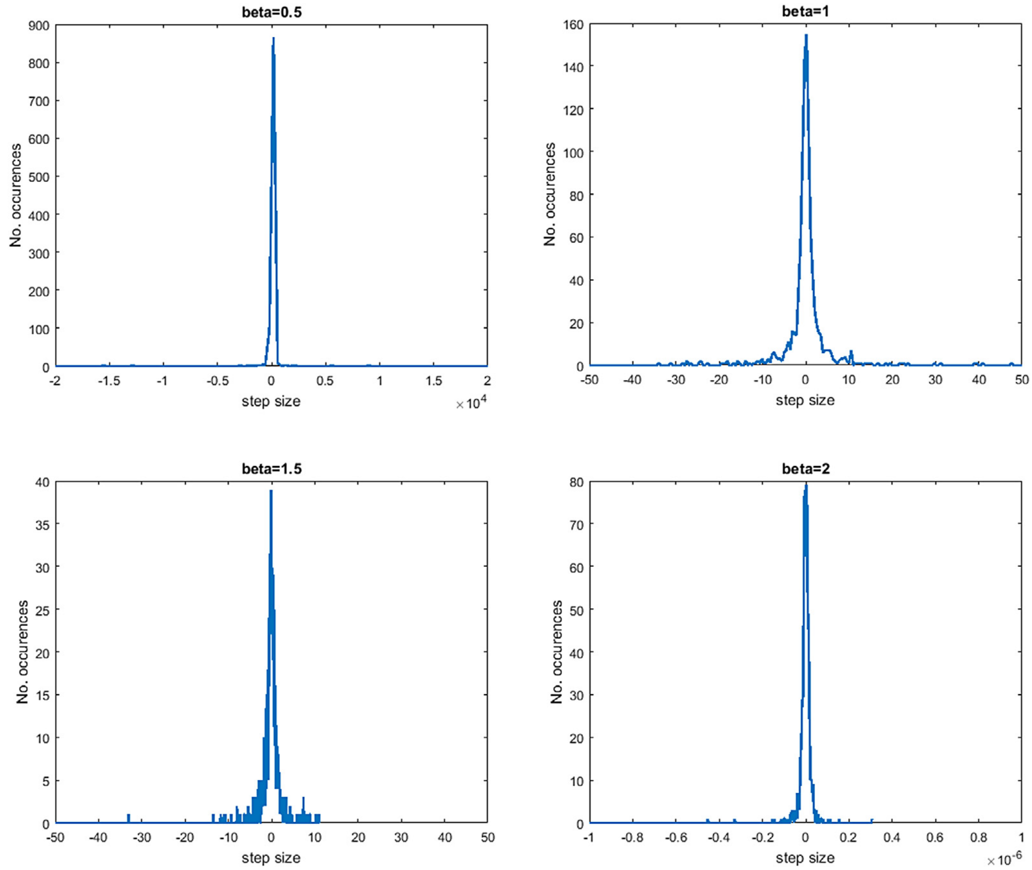
Func.	$\beta = 0.5$				$\beta = 2$			
	LWOA	WOA	LSCA	SCA	LWOA	WOA	LSCA	SCA
F01	**0.0**	**0.0**	64 707	**12.7**	**0.0**	**0.0**	256.7	**12.7**
F02	**0.0**	**0.0**	52.783	**0.010**	**0.0**	**0.0**	0.488	**0.010**
F03	**50.5e+3**	52.3e+3	95 671	**9629**	74.8e+3	**52.3e+3**	12 774	**9629**
F04	**11.605**	19.849	82.994	**19.925**	**0.083**	19.849	**17.052**	19.925
F05	**27.728**	28.103	23.8e+7	**0.0**	28.667	**28.103**	39.0e+4	**3.4e+4**
F06	**0.080**	0.386	62 770	**16.8**	0.581	**0.386**	278.3	**16.8**
F07	0.004	**0.001**	**0.003**	0.035	**0.0**	0.001	**0.011**	0.035
F08	−10.1e+3	**−10.4e+3**	**−4616.8**	−3799.9	−9345.3	**−10 378.9**	−3699.7	**−3799.9**
F09	11.466	**0.0**	293.2	**40.1**	**0.0**	**0.0**	105.6	**40.1**
F10	**0.0**	**0.0**	20.021	**10.869**	**0.0**	**0.0**	15.240	**10.869**
F11	0.039	**0.028**	585.1	**0.9**	**0.0**	0.028	3.592	**0.874**
F12	**0.004**	0.029	51.3e+7	**0.0**	**0.028**	0.029	33.8e+4	**10.9e+4**
F13	**0.087**	0.489	11.9e+8	**0.0**	**0.436**	0.489	58.9e+4	**6.1e+4**
F14	**0.998**	3.015	**1.148**	1.594	**2.915**	3.015	2.688	**1.594**
F15	0.001	**0.001**	0.001	**0.001**	0.001	**0.001**	**0.001**	**0.001**
F16	**−1.032**	**−1.032**	−1.031	**−1.032**	**−1.032**	**−1.032**	−1.032	**−1.032**
F17	**0.398**	0.398	**0.399**	0.400	0.398	**0.398**	0.401	**0.400**
F18	**3.0**	**3.0**	3.001	**3.0**	**3.0**	**3.0**	**3.0**	**3.0**
F19	**−3.860**	−3.850	**−3.855**	−3.853	−3.845	**−3.850**	**−3.855**	−3.853
F20	**−3.280**	−3.271	**−3.070**	−3.057	−3.257	**−3.271**	**−3.059**	−3.057
F21	**−7.858**	−7.793	−2.282	**−2.669**	−7.110	**−7.793**	**−2.793**	−2.669
F22	−7.744	**−7.984**	**−3.486**	−2.709	−6.819	**−7.984**	**−3.675**	−2.709
F23	−6.867	**−7.393**	**−4.414**	−4.258	−6.837	**−7.393**	**−4.283**	−4.258
F24	**90.2**	140.9	169.7	**137.7**	165.7	**140.9**	152.3	**137.7**
F25	**99.1**	182.3	120.0	**115.8**	**168.6**	182.3	123.0	**115.8**
F26	**281.3**	411.7	**407.0**	424.6	448.4	**411.7**	437.7	**424.6**
F27	**442.1**	545.0	465.8	**452.0**	599.0	**545.0**	**448.0**	452.0
F28	**112.2**	209.3	152.9	**130.6**	**161.8**	209.3	136.7	**130.6**
F29	**571.3**	754.9	580.4	**578.6**	792.5	**754.9**	620.6	**578.6**

where Γ is the Gamma function. The value of the parameter β determines the shape of the Levy probability density function, especially in the tail region.

Fig. 2 shows a random sample walking using Lèvy distribution.

Lèvy flights are more efficient than Brownian random walks in exploring unknown, large-scale search space which can be remarked from the large abrupt jumps in the walk. Mathematically this can be interpreted by the fact that the variance increases much faster than the linear relationship of the Brownian random walk. Such interesting properties in Lèvy motivate exploiting such walk for enhancing the exploration capabilities of the adopted optimizers.

Table 5
The shape of random distribution at the different β values.



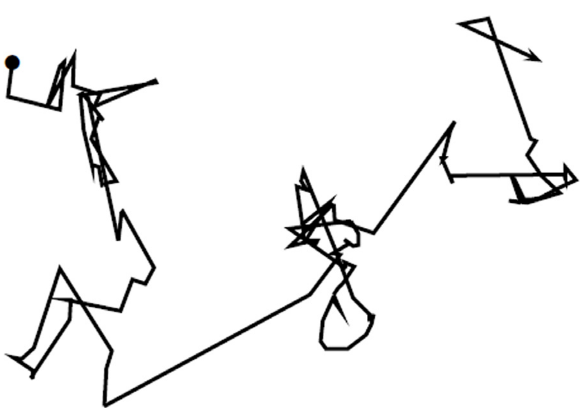


Fig. 2. Lèvy flights in consecutive 50 steps starting at the origin; after [19].

In brief, there are some interesting properties in the adoption of Lèvy flight in random walking especially in the exploration phase which can be summarized as:

- Lèvy flight has the ability to generate a fraction of solutions far enough from the current best solution allowing it to escape from local optima and avoid stagnation.
- Lèvy flights are more efficient than Brownian random walks in exploring unknown, large-scale search space which can be remarked from the large abrupt jumps in the walk.

The above mentioned interesting properties motivate adopting Lèvy random walk in modern optimization methods as a replacement for the exploration phases used in such optimizers.

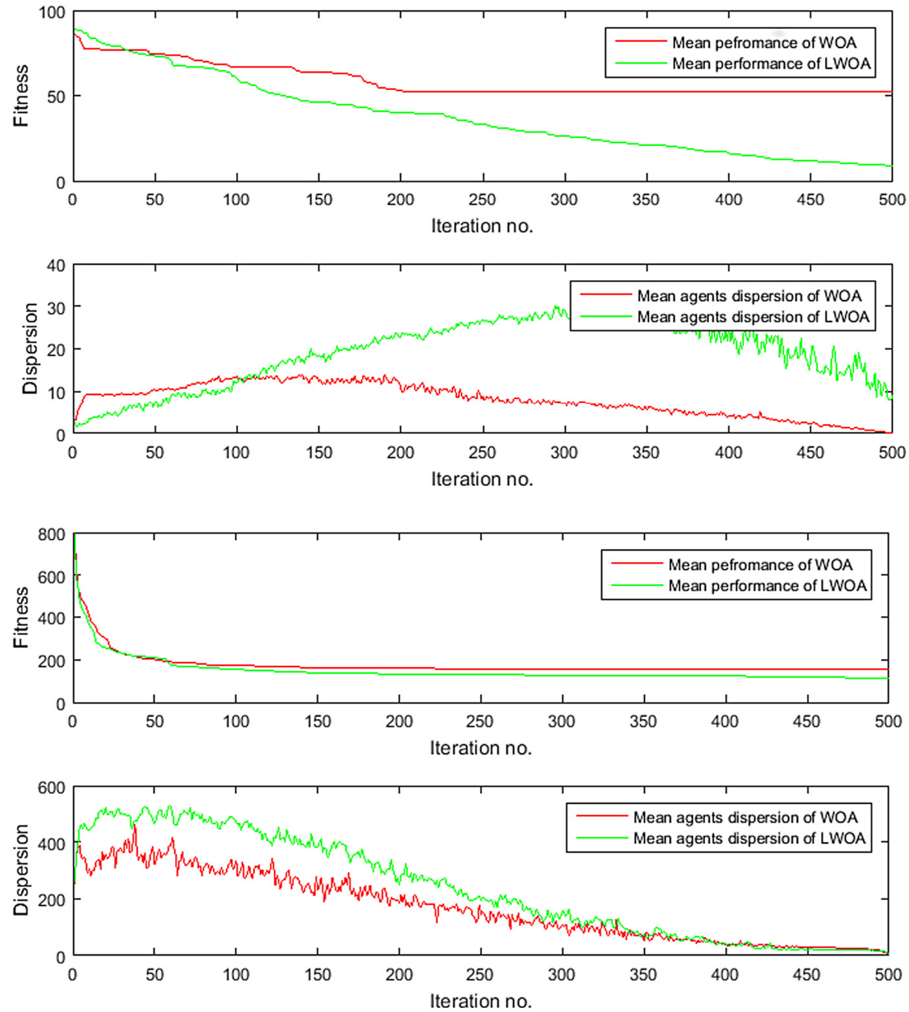
Although some studies try to exploit such interesting properties of Lèvy flights, we can remark that the study in [12] uses that walk to update the positions of some agents who are in terrible — in local minima. But, that proposed variant spends a lot of execution time to classify the search agents as either in local minima or not also the algorithm needs an extra space to store the history of fitness for each agent to make such classification. The study proposed in [13] proposed applying Lèvy flights as a walk following the original walk resulted from WOA random walk. Therefore, all agents change their positions according to the random walk of WOA then update their positions according to the Lèvy flights. Besides that there is a huge amount of execution time spent for such double walking, there is no clear reason to abandon the position of WOA by applying Lèvy to all agents with no intermediate fitness evaluation.

In the study on hand, Lèvy flights were adopted fairly to all search agents on all dimensions with no extra memory storage or extra execution time.

3.1. Lè vy Whale Optimization Algorithm (LWOA)

According to the native WOA, the WOA consists of three main behaviors namely *Encircling prey*, *bubble-net attacking* and *search for prey*. Most of the exploitation activity of the search agents is performed in the bubble-net attacking where the search agents update their own positions according to the current leader position (prey location) using the *spiral equation* to mimic the helix-shaped movement of humpback whales. Basically, in the exploration phase of WOA, A spiral function connecting the current best -*best*- and

Table 6
Convergence curve and agents dispersion for WOA and LWOA on F4, F25.



the search agent to be repositioned -X_i- is constructed and the new location for X_i will be placed at a random location on the spiral curve. The random positioning follows uniform distribution and hence every point on the spiral curve is a candidate to be the new location for -X_i. So, every search agent -X_i has the same chance to be repositioned between the *best* and X_i or beyond X_i- see Fig. 3. On the long run and using uniform distribution, half the search agents will be attracted by the current best – exploitative agents – while the remaining agents will be far from the current best — explorative agents. Such behavior of search consumes many iterations in the exploration even at the end stages of the optimization while more time should be adopted for exploitation, also at the begin of optimization it consumes a lot of time exploiting around the best solution while it should be more explorative at this time.

The above-mentioned properties of the spiral walk using uniform distribution and the need to adopt the interesting properties of Lèvy flight presented in the past section motivates replacing the spiral walk with Lèvy flight in order to keep the interesting properties of that walk. Using Lèvy flight as a random walk, a fraction of search agents will be placed apart from the current best allowing for exploration while the majority of search agents will be positioned between the current best and the original positions of such agents and hence better balance between exploration and exploitation is presented.

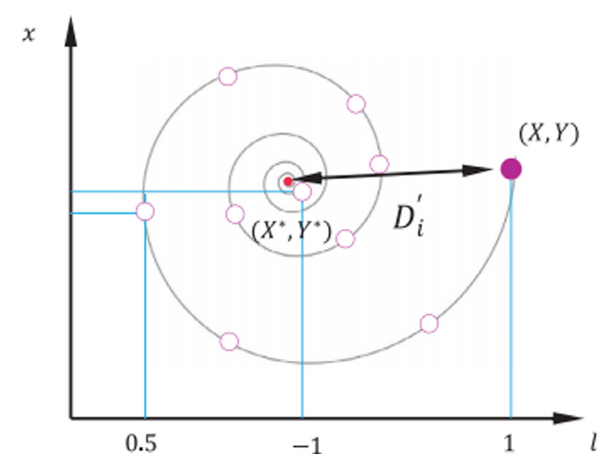


Fig. 3. Spiral motion of the whale search algorithm bubble-net attacking behavior [17].

Therefore, the main position updating equation for the proposed LWOA can be stated as follows:

Table 7
Mean performance of all optimizers adopted in the study at $\beta = 1$ for Lèvy distribution.

Func.	LWOA	WOAL	WOA	CS	PSO	SCA	SCAL	LSCA
F01	**0.0**	**0.0**	**0.0**	987.9	65 871	12.7	30.3	55 290
F02	**0.0**	**0.0**	**0.0**	38.6	246.2	**0.0**	**0.0**	**0.0**
F03	24 429	**7856**	52 272	14 032	417 309	9629	8681	78 560
F04	**1.9**	11.3	19.8	24.9	141.8	19.9	33.4	85.6
F05	**25.1**	25.2	28.1	114 648	416 422 158	33 589	101 471	65 323 990
F06	**0.0**	**0.0**	0.4	1181	62 800	16.8	17.1	52 788
F07	**0.0**	**0.0**	**0.0**	0.2	216.4	**0.0**	0.2	**0.0**
F08	−9244.0	−9373.7	−10 378.9	−6066.6	**−21 355.3**	−3799.9	−3845.3	−4546.1
F09	**0.0**	15.6	**0.0**	195.0	491.4	40.1	42.4	109.2
F10	**0.0**	**0.0**	**0.0**	18.442	20.069	10.869	16.322	9.260
F11	**0.0**	**0.0**	**0.0**	11.3	619.8	0.9	1.2	599.1
F12	**0.0**	**0.0**	**0.0**	25.2	1 371 966 713	109 128	29 180	401 135 246
F13	0.1	**0.0**	0.5	23 014	3 062 527 078	61 494	89 728	864 681 547
F14	**0.998**	1.982	3.015	**0.998**	9.163	1.594	2.480	1.049
F15	**0.001**	**0.001**	**0.001**	**0.001**	0.004	**0.001**	**0.001**	**0.001**
F16	−**1.032**	−**1.032**	−**1.032**	−**1.032**	−**1.032**	−**1.032**	−**1.032**	−**1.032**
F17	**0.398**	**0.398**	**0.398**	**0.398**	**0.398**	0.400	0.400	0.399
F18	**3.0**	**3.0**	**3.0**	**3.0**	11.100	**3.0**	**3.0**	**3.0**
F19	−3.852	−3.862	−3.850	−3.782	−**3.863**	−3.853	−3.854	−3.854
F20	−**3.307**	−3.268	−3.271	−3.305	−3.208	−3.057	−2.761	−3.068
F21	−6.717	−7.107	−7.793	−**10.153**	−4.499	−2.669	−3.064	−2.826
F22	−7.341	−7.677	−**7.984**	−7.153	−5.048	−2.709	−3.618	−3.516
F23	−6.466	−6.080	−**7.393**	−6.453	−4.915	−4.258	−3.880	−4.639
F24	106.3	120.2	140.9	**6.2**	330.1	137.7	147.6	133.4
F25	140.7	133.5	182.3	**35.7**	343.5	115.8	111.1	114.6
F26	321.1	363.7	411.7	**248.2**	513.8	424.6	397.0	377.1
F27	496.2	559.8	545.0	**369.8**	649.5	452.0	456.7	427.3
F28	90.7	**86.1**	209.3	93.0	516.1	130.6	133.3	129.5
F29	716.8	779.6	754.9	726.2	897.4	578.6	610.7	**550.7**

Table 8
Mean performance of all optimizers adopted in the study at $\beta = 1.5$ for Lèvy distribution.

Func.	LWOA	WOAL	WOA	CS	PSO	SCA	SCAL	LSCA
F01	**0.0**	**0.0**	**0.0**	979.5	65 871	12.7	10.9	19 962
F02	**0.0**	**0.0**	**0.0**	30.7	246.2	**0.0**	**0.0**	**0.0**
F03	14 638	20 339	52 272	13 005	417 309	**9629**	10 155	61 069
F04	**0.9**	21.8	19.8	26.3	141.8	19.9	35.6	83.8
F05	26.9	**26.0**	28.1	37 839	416 422 158	33 589	563 426	4 394 313
F06	**0.0**	**0.0**	0.4	957.8	62 800	16.8	14.6	24 337
F07	**0.0**	**0.0**	**0.0**	0.2	216.4	**0.0**	0.1	**0.0**
F08	−9801.1	−9295.5	−10 378.9	−6093.8	**−21 355.3**	−3799.9	−3671.7	−4581.9
F09	**0.0**	13.7	**0.0**	178.3	491.4	40.1	46.4	76.6
F10	**0.0**	**0.0**	**0.0**	13.187	20.069	10.869	13.480	8.945
F11	**0.0**	**0.0**	**0.0**	10.8	619.8	0.9	1.3	575.5
F12	0.1	0.2	**0.0**	19.9	1 371 966 713	109 128	49 071	58 416 226
F13	**0.2**	**0.2**	0.5	22 503	3 062 527 078	61 494	315 722	172 755 818
F14	**0.998**	2.173	3.015	**0.998**	9.163	1.594	2.373	0.998
F15	0.001	**0.0**	0.001	0.001	0.004	0.001	0.001	0.001
F16	−**1.032**	−**1.032**	−**1.032**	−**1.032**	−**1.032**	−**1.032**	−**1.032**	−**1.032**
F17	**0.398**	**0.398**	**0.398**	**0.398**	**0.398**	0.400	0.400	0.399
F18	**3.0**	**3.0**	**3.0**	**3.0**	11.100	**3.0**	**3.0**	**3.0**
F19	−3.853	−3.863	−3.850	−3.852	−**3.863**	−3.853	−3.854	−3.854
F20	−**3.296**	−3.256	−3.271	−3.222	−3.208	−3.057	−2.803	−3.062
F21	−7.085	−**7.993**	−7.793	−7.153	−4.499	−2.669	−1.965	−3.321
F22	−8.272	−7.051	−7.984	−**10.390**	−5.048	−2.709	−4.016	−4.183
F23	−7.159	−6.886	−7.393	−**10.528**	−4.915	−4.258	−3.585	−4.568
F24	75.6	125.6	140.9	**14.6**	330.1	137.7	149.4	144.4
F25	199.8	180.0	182.3	**46.6**	343.5	115.8	112.7	111.5
F26	385.7	463.0	411.7	391.0	513.8	424.6	423.3	**370.4**
F27	564.0	573.1	545.0	**375.0**	649.5	452.0	442.0	439.1
F28	131.1	181.0	209.3	124.5	516.1	130.6	**120.8**	126.1
F29	752.2	800.0	754.9	**513.5**	897.4	578.6	670.4	566.4

$$X^{t+1} = \begin{cases} X_*^t - \overrightarrow{A}\,.\,\overrightarrow{D} & \text{if } p < 0.5 \\ s.\dfrac{u}{|v|^{\frac{1}{\beta}}}.(X^t - leader^t) + X^t & \text{if } p \geq 0.5, \end{cases} \tag{23}$$

where p is a random number in [0, 1], $Leader^t$ is the current best solution at iteration t, X_*^t is set to current best solution at exploitation iterations and is set to a random position of a search agent at exploration iterations, u, v are random numbers according to specification in Eq. (20), β is a constant controlling the Lèvy distribution shape and s is the scale parameter and is adaptively set according to the following linear equation:

$$s = \frac{T - t}{T} * S, \tag{24}$$

Table 9
Mean performance of all optimizers adopted in the study at $\beta = 0.5$ for Lèvy distribution.

Func.	LWOA	WOAL	WOA	CS	PSO	SCA	SCAL	LSCA
F01	**0.0**	0.1	**0.0**	1077	65 871	12.7	89.2	64 707
F02	**0.0**	**0.0**	**0.0**	61.4	246.2	**0.0**	0.2	52.8
F03	50 543	15 825	52 272	14 867	417 309	**9629**	12 132	95 671
F04	**11.6**	14.4	19.8	24.6	141.8	19.9	36.6	83.0
F05	**27.7**	382.4	28.1	201 328	416 422 158	33 589	163 928	238 420 324
F06	**0.1**	1.5	0.4	1068	62 800	16.8	133.5	62 770
F07	**0.0**	0.1	**0.0**	0.2	216.4	**0.0**	0.3	**0.0**
F08	−10 095.4	−9210.4	−10 378.9	−6359.7	**−21 355.3**	−3799.9	−3855.2	−4616.8
F09	11.5	70.4	**0.0**	238.1	491.4	40.1	51.3	293.2
F10	**0.0**	0.111	**0.0**	19.550	20.069	10.869	17.404	20.021
F11	**0.0**	0.3	**0.0**	10.8	619.8	0.9	1.8	585.1
F12	**0.0**	0.5	**0.0**	126.3	1 371 966 713	109 128	99 587	513 298 162
F13	**0.1**	0.7	0.5	30 515	3 062 527 078	61 494	1 078 052	1 189 182 677
F14	**0.998**	**0.998**	3.015	**0.998**	9.163	1.594	1.312	1.148
F15	**0.001**	**0.001**	**0.001**	**0.001**	0.004	**0.001**	**0.001**	**0.001**
F16	**−1.032**	**−1.032**	**−1.032**	**−1.032**	**−1.032**	**−1.032**	**−1.032**	−1.031
F17	**0.398**	**0.398**	**0.398**	**0.398**	**0.398**	0.400	0.400	0.399
F18	**3.0**	**3.0**	**3.0**	**3.0**	11.100	**3.0**	**3.0**	3.001
F19	−3.860	−3.861	−3.850	−3.855	**−3.863**	−3.853	−3.855	−3.855
F20	**−3.280**	−3.199	−3.271	−3.278	−3.208	−3.057	−3.032	−3.070
F21	**−7.858**	−6.364	−7.793	−7.153	−4.499	−2.669	−2.986	−2.282
F22	−7.744	−6.880	**−7.984**	−7.403	−5.048	−2.709	−3.196	−3.486
F23	−6.867	−5.021	−7.393	**−10.536**	−4.915	−4.258	−3.492	−4.414
F24	90.2	95.0	140.9	**5.1**	330.1	137.7	149.6	169.7
F25	99.1	125.5	182.3	**45.8**	343.5	115.8	114.9	120.0
F26	281.3	298.6	411.7	**265.4**	513.8	424.6	372.0	407.0
F27	442.1	519.1	545.0	**367.2**	649.5	452.0	446.6	465.8
F28	112.2	83.4	209.3	**20.2**	516.1	130.6	136.6	152.9
F29	571.3	688.4	754.9	**515.8**	897.4	578.6	615.7	580.4

Table 10
Mean performance of all optimizers adopted in the study at $\beta = 2$ for Lèvy distribution.

Func.	LWOA	WOAL	WOA	CS	PSO	SCA	SCAL	LSCA
F01	**0.0**	**0.0**	**0.0**	1033	65 871	12.7	15.7	256.7
F02	**0.0**	**0.0**	**0.0**	56.9	246.2	**0.0**	**0.0**	0.5
F03	74 756	42 345	52 272	11 894	417 309	9629	**8445**	12 774
F04	**0.1**	56.7	19.8	25.6	141.8	19.9	32.8	17.1
F05	28.7	**27.9**	28.1	203 342	416 422 158	33 589	76 068	389 576
F06	0.6	**0.4**	**0.4**	1080	62 800	16.8	22.9	278.3
F07	**0.0**	**0.0**	**0.0**	0.4	216.4	**0.0**	0.1	**0.0**
F08	−9345.3	−10 269.9	−10 378.9	−4946.8	**−21 355.3**	−3799.9	−3910.7	−3699.7
F09	**0.0**	**0.0**	**0.0**	227.9	491.4	40.1	30.1	105.6
F10	**0.0**	**0.0**	**0.0**	14.934	20.069	10.869	14.837	15.240
F11	**0.0**	**0.0**	**0.0**	10.7	619.8	0.9	1.2	3.6
F12	**0.0**	**0.0**	**0.0**	27.6	1 371 966 713	109 128	32 471	338 439
F13	**0.4**	0.5	0.5	44 202	3 062 527 078	61 494	73 843	589 234
F14	2.915	3.257	3.015	**0.998**	9.163	1.594	1.987	2.688
F15	**0.001**	**0.001**	**0.001**	**0.001**	0.004	**0.001**	**0.001**	**0.001**
F16	**−1.032**	**−1.032**	**−1.032**	**−1.032**	**−1.032**	**−1.032**	**−1.032**	**−1.032**
F17	**0.398**	**0.398**	**0.398**	**0.398**	**0.398**	0.400	0.400	0.401
F18	**3.0**	**3.0**	**3.0**	**3.0**	11.100	**3.0**	**3.0**	**3.0**
F19	−3.845	−3.852	−3.850	−3.850	**−3.863**	−3.853	−3.855	−3.855
F20	−3.257	−3.265	**−3.271**	−3.219	−3.208	−3.057	−2.842	−3.059
F21	−7.110	−7.723	**−7.793**	−7.029	−4.499	−2.669	−2.667	−2.793
F22	−6.819	−7.075	**−7.984**	−6.276	−5.048	−2.709	−3.040	−3.675
F23	−6.837	−6.743	−7.393	**−10.367**	−4.915	−4.258	−4.036	−4.283
F24	165.7	154.1	140.9	**67.4**	330.1	137.7	136.3	152.3
F25	168.6	195.0	182.3	**72.8**	343.5	115.8	115.9	123.0
F26	448.4	528.3	411.7	**330.7**	513.8	424.6	388.4	437.7
F27	599.0	566.6	545.0	**400.3**	649.5	452.0	455.7	448.0
F28	161.8	196.8	209.3	130.5	516.1	130.6	**129.5**	136.7
F29	792.5	799.7	754.9	**525.6**	897.4	578.6	580.7	620.6

where T is the number of iterations to run the optimization, t is the current iteration numbers and S is set to 0.1*range [19] with $range = ub - lb$ and ub, lb are the upper and lower limit of the problem space.

3.2. Lèvy Sine Cosine Algorithm (LSCA)

Mainly, the SCA performs its walk by encircling the current best solution using sine and cosine functions as mentioned in Eq. (2). A given search agent switches between sine and cosine position

Table 11
Mean performance of all optimizers adopted in the study at $\beta = 1$ for Lèvy distribution — standard deviation of performance.

Func.	LWOA	WOAL	WOA	CS	PSO	SCA	SCAL	LSCA
F01	**0.0**	**0.0**	**0.0**	204.8	28 347	23.8	47.8	11 697
F02	**0.0**	**0.0**	**0.0**	6.789	47.937	0.011	0.069	0.006
F03	9581	3626	18 065	**2312**	202 286	6663	4288	10 939
F04	**1.739**	8.746	17.858	1.893	15.131	12.761	11.475	2.234
F05	**0.1**	0.6	0.5	30 965	232 641 508	65 022	235 784	53 865 071
F06	**0.0**	**0.0**	0.3	269.7	17 083	16.6	17.2	15 752
F07	**0.0**	**0.0**	**0.0**	**0.0**	126.1	**0.0**	0.2	**0.0**
F08	877.0	1207	1728	424.2	3363	**237.3**	390.7	358.4
F09	**0.0**	21.2	**0.0**	13.2	113.2	36.3	34.2	45.3
F10	**0.0**	**0.0**	**0.0**	1.094	0.072	9.935	6.812	2.117
F11	0.1	**0.0**	0.1	2.2	186.5	0.4	0.4	105.1
F12	**0.0**	**0.0**	**0.0**	5.5	8 409 036 14	457 496	73 780	165 307 102
F13	**0.1**	**0.1**	0.3	35 075	1 777 052 426	183 967	256 156	264 031 035
F14	**0.0**	2.244	2.895	**0.0**	5.723	0.932	2.186	0.222
F15	**0.0**	**0.0**	**0.0**	**0.0**	0.009	**0.0**	**0.0**	**0.0**
F16	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
F17	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	0.002	0.002	0.001
F18	**0.0**	**0.0**	**0.0**	**0.0**	24.931	**0.0**	**0.0**	**0.0**
F19	0.013	0.003	0.016	0.128	**0.0**	0.003	0.002	0.004
F20	0.047	0.061	0.081	**0.036**	0.286	0.040	0.470	0.066
F21	2.639	2.915	2.545	**0.0**	2.672	2.019	2.103	1.957
F22	2.866	2.812	3.121	2.510	3.253	2.062	**1.737**	1.862
F23	3.217	3.557	3.321	**0.062**	3.442	1.017	1.523	0.856
F24	118.4	132.1	111.3	**8.6**	213.9	49.3	38.8	24.4
F25	91.6	107.3	108.7	26.0	153.4	**7.1**	**7.1**	25.7
F26	125.7	140.7	134.7	**23.3**	191.6	118.3	50.7	60.0
F27	128.6	149.8	119.8	22.7	183.9	36.6	**21.6**	24.1
F28	93.2	91.3	157.6	**10.3**	174.9	63.7	60.0	59.8
F29	206.5	184.6	200.3	**68.7**	92.1	132.9	146.7	135.0

Table 12
Mean performance of all optimizers adopted in the study at $\beta = 1.5$ for Lèvy distribution — standard deviation of performance.

Func.	LWOA	WOAL	WOA	CS	PSO	SCA	SCAL	LSCA
F01	**0.0**	**0.0**	**0.0**	142.5	28 347	23.8	15.1	6154
F02	**0.0**	**0.0**	**0.0**	15.998	47.937	0.011	0.053	0.006
F03	7581	6069	18 065	**1967**	202 286	6663	9483	11 289
F04	**0.796**	20.475	17.858	1.495	15.131	12.761	13.573	4.954
F05	0.6	0.8	**0.5**	15 538	232 641 508	65 022	2 350 212	5 049 638
F06	**0.1**	**0.1**	0.3	253.1	17 083	16.6	18.0	12 053
F07	**0.0**	**0.0**	**0.0**	0.1	126.1	**0.0**	0.1	**0.0**
F08	1521	1348	1728	383.6	3363	**237.3**	278.5	270.5
F09	**0.0**	30.3	**0.0**	14.3	113.2	36.3	35.2	29.8
F10	**0.0**	**0.0**	**0.0**	1.449	0.072	9.935	8.653	4.984
F11	**0.0**	**0.0**	0.1	1.5	186.5	0.4	0.9	86.9
F12	0.3	0.7	**0.0**	4.0	840 903 614	457 496	147 578	78 254 798
F13	**0.2**	**0.2**	0.3	36 293	1 777 052 426	183 967	1 032 670	171 189 633
F14	**0.0**	3.0	2.895	**0.0**	5.723	0.932	2.979	**0.0**
F15	**0.0**	**0.0**	**0.0**	**0.0**	0.009	**0.0**	**0.0**	**0.0**
F16	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
F17	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	0.002	0.002	0.002
F18	0.001	**0.0**	**0.0**	**0.0**	24.931	**0.0**	**0.0**	**0.0**
F19	0.009	**0.0**	0.016	**0.0**	**0.0**	0.003	0.004	0.005
F20	0.045	0.061	0.081	**0.0**	0.286	0.040	0.378	0.032
F21	2.551	2.764	2.545	**0.001**	2.672	2.019	2.017	1.637
F22	2.668	2.897	3.121	**0.035**	3.253	2.062	1.684	1.127
F23	2.848	2.812	3.321	**0.013**	3.442	1.017	1.961	0.631
F24	73.7	124.6	111.3	**23.9**	213.9	49.3	42.5	53.4
F25	92.0	114.1	108.7	32.1	153.4	**7.1**	8.5	10.1
F26	105.6	172.2	134.7	**17.9**	191.6	118.3	86.8	60.0
F27	147.0	124.6	119.8	**13.2**	183.9	36.6	20.5	16.6
F28	89.0	145.8	157.6	**14.9**	174.9	63.7	57.8	56.5
F29	181.1	176.3	200.3	**6.2**	92.1	132.9	191.6	128.0

updating randomly while it applies exploration or exploitation according to random scale factors; $r2$ in Eq. (25). Assuming sinusoidal function and uniform distribution, any given point on the cosine function has the chance to be a candidate new location for a given search agent X_i. On the long run and at any given time, half the search agents will fall between their current position and the current global best – exploiting the global best – while the remaining agents will fall far away from the current best beyond their current location — exploring the space. Such fair fractionizing of search agents between exploration and exploitation again can consume a lot of time in exploration at the end stages of optimization while it should devote more time to the exploitation while on the other hand, it should devote more time for exploration at the begin of optimization while actually, many search agents will work on exploitation.

Table 13
Mean performance of all optimizers adopted in the study at $\beta = 0.5$ for Lèvy distribution — standard deviation of performance.

Func.	LWOA	WOAL	WOA	CS	PSO	SCA	SCAL	LSCA
F01	**0.0**	0.1	**0.0**	260.0	28 347	23.8	205.4	12 258
F02	**0.0**	0.021	**0.0**	6.183	47.937	0.011	0.212	8.844
F03	11 691	4727	18 065	**2196**	202 286	6663	6091	12 967
F04	6.559	5.853	17.858	**2.081**	15.131	12.761	11.280	2.453
F05	**0.3**	547.3	0.5	142 326	232 641 508	65 022	219 679	50 811 937
F06	**0.0**	0.5	0.3	157.7	17 083	16.6	184.9	7442
F07	**0.0**	0.1	**0.0**	**0.0**	126.1	**0.0**	0.4	**0.0**
F08	1334	1054	1728	**167.2**	3363	237.3	313.1	216.7
F09	26.7	21.5	**0.0**	12.4	113.2	36.3	25.8	47.5
F10	**0.0**	0.070	**0.0**	0.489	0.072	9.935	6.947	0.289
F11	**0.1**	**0.1**	**0.1**	2.4	186.5	0.4	1.3	99.9
F12	**0.0**	0.5	**0.0**	249.8	840 903 614	457 496	196 272	148 863 504
F13	**0.0**	0.3	0.3	24 114	1 777 052 426	183 967	2 368 211	164 435 494
F14	**0.0**	**0.0**	2.895	**0.0**	5.723	0.932	0.641	0.413
F15	**0.0**	**0.0**	**0.0**	**0.0**	0.009	**0.0**	**0.0**	**0.0**
F16	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
F17	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	0.002	0.003	0.001
F18	**0.0**	**0.0**	**0.0**	**0.0**	24.931	**0.0**	**0.0**	0.001
F19	0.003	0.004	0.016	0.014	**0.0**	0.003	0.004	0.004
F20	0.059	0.087	0.081	0.060	0.286	**0.040**	0.048	0.044
F21	2.601	3.002	2.545	**0.0**	2.672	2.019	1.873	1.636
F22	2.726	3.379	3.121	**0.0**	3.253	2.062	1.697	1.789
F23	3.568	3.049	3.321	**0.001**	3.442	1.017	1.606	0.575
F24	102.1	88.7	111.3	**5.0**	213.9	49.3	30.2	24.2
F25	91.2	85.9	108.7	**5.1**	153.4	7.1	8.5	9.6
F26	99.5	118.3	134.7	**33.8**	191.6	118.3	65.8	58.7
F27	134.8	154.2	119.8	**10.8**	183.9	36.6	25.3	51.3
F28	105.9	116.3	157.6	**4.2**	174.9	63.7	66.6	67.1
F29	162.8	213.8	200.3	**10.1**	92.1	132.9	175.2	140.7

Table 14
Mean performance of all optimizers adopted in the study at $\beta = 2$ for Lèvy distribution — standard deviation of performance.

Func.	LWOA	WOAL	WOA	CS	PSO	SCA	SCAL	LSCA
F01	**0.0**	**0.0**	**0.0**	270.7	28 347	23.8	28.4	275.0
F02	**0.0**	**0.0**	**0.0**	11.334	47.937	0.011	0.060	0.437
F03	19 186	18 437	18 065	**1517**	202 286	6663	5471	6895
F04	**0.115**	30.513	17.858	1.669	15.131	12.761	11.822	12.718
F05	**0.1**	0.5	0.5	63 877	232 641 508	65 022	235 935	546 010
F06	**0.2**	0.2	0.3	117.8	17 083	16.6	36.3	435.9
F07	**0.0**	**0.0**	**0.0**	0.1	126.1	**0.0**	0.3	**0.0**
F08	1418	1794	1728	**156.2**	3363	237.3	377.5	271.8
F09	**0.0**	**0.0**	**0.0**	7.7	113.2	36.3	28.0	52.6
F10	**0.0**	**0.0**	**0.0**	1.533	0.072	9.935	8.594	7.115
F11	**0.0**	**0.0**	0.1	1.4	186.5	0.4	0.9	3.5
F12	**0.0**	**0.0**	**0.0**	5.8	840 903 614	457 496	108 462	502 361
F13	**0.2**	0.3	0.3	34 259	1 777 052 426	183 967	226 429	705 870
F14	3.067	3.353	2.895	**0.0**	5.723	0.932	2.242	2.914
F15	0.001	0.001	**0.0**	**0.0**	0.009	**0.0**	**0.0**	**0.0**
F16	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
F17	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	0.002	0.002	0.003
F18	0.001	**0.0**	**0.0**	**0.0**	24.931	**0.0**	0.001	0.001
F19	0.045	0.028	0.016	**0.0**	**0.0**	0.003	0.004	0.003
F20	0.072	0.094	0.081	**0.001**	0.286	0.040	0.381	0.045
F21	2.363	2.778	2.545	**0.013**	2.672	2.019	1.618	1.850
F22	2.680	3.207	3.121	**0.028**	3.253	2.062	2.103	1.172
F23	3.242	3.333	3.321	**0.223**	3.442	1.017	1.382	1.403
F24	104.0	133.6	111.3	**18.0**	213.9	49.3	30.3	25.9
F25	80.8	115.0	108.7	8.3	153.4	**7.1**	9.6	29.0
F26	103.3	175.5	134.7	**24.2**	191.6	118.3	59.1	55.0
F27	143.8	132.0	119.8	**19.8**	183.9	36.6	49.9	49.2
F28	70.4	130.5	157.6	**12.3**	174.9	63.7	60.1	57.9
F29	179.2	166.9	200.3	**5.0**	92.1	132.9	161.2	161.7

Again, Lèvy random walk with the interesting properties mentioned in the above section can be adopted as a replacement for the exploration phase so that only a fraction of search agents will be devoted to exploration even at the end stages of optimization while the remaining search agents will work on exploitation. The final position update equation is formulated as:

$$x_{ij}^{t+1} = \begin{cases} x_{ij}^t + r_1 * sin(r_2) * |r_3 best_j^t - x_{ij}^t| \text{ if } r < 0.5 \\ s.\dfrac{u}{|v|^{\frac{1}{\beta}}}.(x_{ij}^t - best_j^t) + x_{ij}^t \text{ if } r \geq 0.5, \end{cases} \tag{25}$$

where r is a random number drawn from uniform distribution in the range $[0, 1]$, x_{ij}^t is the position of solution i at dimension j at iteration t, $best_j^t$ is the current best solution at iteration t in dimension j, r_1 is a control parameter that balance the exploration and exploitation phases of the algorithm; see Eq. (3), r_2 defines how far the movement should be from the destination, r_3 is a random number controlling the effect of the destination to the new position, s is scale factor defined according to Eq. (24), u, v are random numbers according to specification in Eq. (20) and β is a constant controlling the Lèvy distribution shape.

4. Experimental results and discussions

This section introduces and describes the set of examined dataset benchmark test functions, adopted optimizers, evaluation criteria, the numerical results, and discussion.

4.1. Data description

This work applies and compares the same examined datasets in [17] consisting of a total of 29 benchmark functions of two subsets. The first set of functions consists of 23 benchmark functions while the second subset consists of 6 composite functions. The following sections describe the two different sets extensively.

The first 23 functions are standard benchmark problems applied in the optimization review [21–24]. Table 1 represents some of the test functions describing the cost function, the range of variation of optimization variables and the optimal value (F_{min}) cited in the literature [17]. The standard benchmark test functions can be divided into unimodal, multimodal, fixed dimension multimodal, and composite test functions.

The functions F1–F7 are *unimodal* that have one global optimum. These functions provide evaluating the exploitation ability of the examined meta-heuristic methods. The distinction between the multimodal functions and the fixed dimensional multi-modal functions is the capability of determining the coveted problem size (number of design variables).

The fixed dimensional multimodal functions do not permit to tune the number of design variables, but they allow various search space compared to multimodal functions. Moreover, the composite functions provide challenging functions by changing the global optimum to random points before each iteration to discover the global optimum on boundaries of search spaces [25].

The multimodal functions involve multiple local optima whose number increases exponentially with the problem size. Hence, these test functions become very helpful if the objective is to compute the exploration ability of the optimization method [17].

The CEC2005 composite benchmark functions set is developed novel benchmark functions that possess several desirable properties and introduce in the session of function optimization at CEC2005 [26]. The challenge behind the invention of these composite functions is to build a more challenging function with a randomly placed global optimum and many randomly located deep local optima from the standard benchmark functions. Gaussian functions are used to couple these benchmark functions and blur the individual function's structure [26]. The explanation and detailed description of these CEC2005 composite functions are in [27].

As is clear from the selection of the data set used, the proposed algorithm is to be assessed on its continuous form. A discrete version of the algorithm can be designed and applied for discrete and binary problems either by suitable threshold the continuous algorithm stops and steps [28,29] or by considering only the inspiration of the continuous algorithm and designing a full binary/discrete version [30,31]. Both mechanisms can be considered in a future publication.

4.2. Evaluation criteria

In this study, the proposed optimizers are presented to each of 29 mathematical optimization problems for assessing optimizers' numerical efficiency. The two used optimizers were compared against their Lèvy variant over all the optimization functions. Moreover, the proposed optimizers were compared to two common EC methods namely CS [5] and particle swarm optimization(PSO) [32]. CS is adopted for comparison as it uses Lèvy flight as its basic random walk style and hence we intended to clarify the difference in performance for the proposed variant against the native CS algorithm. PSO was adopted for comparison for its popularity as a reference optimizer for performance comparison. Two recent optimizers were proposed in literature adopting Lèvy on WOA [13] and SCA [12] were also adopted in the comparison. We mentioned these two variants of WOA and SCA to highlight that the impact of the Lèvy flight on a given optimizer differs depending on the time and mechanism of applying such a walk.

In the work in [13] Lèvy flights were adopted in WOA to enhance its diversification capability and local optima avoidance. In that study author proposed applying double random walk types to the search agents. All search agents in the swarm update their positions according to the original WOA principles then each agent applies an extra walk by applying Lèvy flights on its updated position. As is clear, such double walking mechanism consumes a huge amount of time and in the same time leads to disrupting the walk style. In the other study in [12], Lèvy flights were adopted to enhance the performance of sine cosine algorithm (SCA). In that study, the authors classify search agents as either in local minima or in nonlocal minima based on the agent's fitness history. The agents in local minima abandon their original walk – using SCA – and uses Lèvy flights as an alternative walk. Again such variant consumes a lot of memory and time to maintain agent's history and to classify agents. Also, as clear from the algorithm Lèvy flights are rarely applied.

For all the algorithms, a population size and maximum iteration equal to 30 and 500 have been utilized. The number of search agents n and number of iteration T are commonly set based on trial and error basis and should be proportional to the size of the search space and the complexity of the fitness function. In this work, we adopted the same setting for n and T as in the original paper by in Mirjalili [17]. The used benchmark functions are the same set used in [33]. It is divided into four groups: unimodal, multimodal, fixed-dimension multimodal, and composite functions to compare different aspects of the optimizer.

The β parameter controlling the Lèvy distribution is set to 1.5 for all experiments [34] and three different settings of β were also adopted to check for parameter dependency at 0.5, 1 and 2. The detailed setting and description of individual parameters were outlined in Table 2.

For each benchmark function, individual optimizer runs N_{runs} times starting from different populations randomly generated and the following seven evaluation indicators are computed per optimizer:

1. *Mean fitness:* is an average value of all the solutions in the final sets obtained by an optimizer in some individual runs [35].
2. *Statistical standard deviation (std):* is used to ensure that the optimizer convergences to the same optimal and ensures repeatability of the results. It is computed over all the sets of final solutions obtained by an optimizer in a number of individual runs [36].
3. *Wilcoxon rank sum test:* assigns a rank to all the scores considered as one group and then sums the ranks of each group [37]. The rank-sum test is often described as the

Table 15
Significance measures for LWOA against the other optimizers adopted in the study.

LWOA	Significance measure	$\beta = 1$	$\beta = 1.5$	$\beta = 0.5$	$\beta = 2$
WOAL	p-value of t-test	0	0.05	0.09	0.08
	p-value of Wilcoxon-test	0.05	0.04	0.087	0.076
WOA	p-value of t-test	0.02	0.01	0.06	0.002
	p-value of Wilcoxon-test	0.039	0.049	0.094	0.091
CS	p-value of t-test	0.041	0.08	0.9	0.07
	p-value of Wilcoxon-test	0.05	0.08	0.1	0.098
PSO	p-value of t-test	0.01	0.01	0.02	0.04
	p-value of Wilcoxon-test	0.05	0.04	0.041	0.05

Table 16
Significance measures for LSCA against the other optimizers adopted in the study.

LSCA	Significance measure	$\beta = 1$	$\beta = 1.5$	$\beta = 0.5$	$\beta = 2$
SCAL	p-value of t-test	0.05	0.01	0.11	0.15
	p-value of Wilcoxon-test	0.033	0.05	0.028	0.09
SCA	p-value of t-test	0.01	0.02	0.09	0.05
	p-value of Wilcoxon-test	0.03	0.03	0.06	0.008
CS	p-value of t-test	0.05	0.04	0.14	0.15
	p-value of Wilcoxon-test	0.002	0.004	0.13	0.09
PSO	p-value of t-test	0.001	0	0.001	0.002
	p-value of Wilcoxon-test	0	0.001	0.003	0.001

nonparametric version of the t test for two independent groups.

4. *T-test:* is statistical significance shows despite whether the contrast between the two groups' midpoints in all possibility mirrors a *real* distinction in the population from the groups were inspected [38].
5. *Run time average:* is the average run time in seconds for an individual optimizer on a given function. The results were conducted on 64-bit windows 10 computer with i7 core 3.6 GHz and 8 GB RAM.

4.3. Numerical results and discussion

Table 3 outlines the average output fitness of WOA and SCA against their proposed Lèvy variants at recommended values for β - 1 and 1.5 across the 30 runs. We can remark from the table that the proposed Lèvy flight variants have a clear enhance over the original algorithms especially at the multi-modal and composite function set. Such enhanced performance can be interpreted by the fact that Lèvy walks provide some steps that are apart from the existing local optima and hence provide the optimizer with the capability to escape from local minima and hence avoid premature convergence. Such remarks can also be noticed from Table 5 where the histogram of the output steps from Lèvy is plotted at the different settings for β. We can see from that figure that histogram looks noisy and some steps values go to infinity which allows for the generation of long jumps in the search space but of course with low probability. A similar interpretation can be derived from the figure in Table 6 where the impact of using Lèvy flights affects the search agents dispersion measure – the standard deviation of agents' fitness – across the optimization time for two objective functions. We can see from the figure that without adopting Lèvy flights WOA converges to local optima and all search agents are around that optima and keeps very low dispersion and hence cannot escape from such optima. On the other hand, we can see that adopting Lèvy in the same situation, the search agents keep higher dispersion measure and hence can much explore around the reached optima and apart from it and hence can escape from such local optima. We can also see that the advance in performance is kept for both SCA and WOA while it is much apparent for WOA as it is much exploitative optimizer which can be remarked from its repositioning equations that is much dependent on the current global best solution.

We can also remark that for unimodal functions there is no clear advance over the original algorithms since the single optima can easily be caught even with less exploration of the search space. which can again be remarked from the figures in Tables 5 and 6.

In case of using much outliers values for β as outlined in Table 4 we can see that such glancing performance is less clear which can be interpreted by the fact that output random steps in these cases become either very small leading to consumption of optimization iterations without reaching the optimum or very large that makes the optimizer oscillating and hence also misses the global optima. The same interpretation can be seen from the figure in Table 5 for β values at 0.5 and 2 we can see either very narrow or very loose histogram of steps.

Tables 7 and 8 outline the comparative average fitness obtained across all the runs for all the adopted optimizers at recommended values for β. Comparing the proposed Lèvy variants for WOA and SCA namely LWOA and LSCA to the Lèvy variant proposed in [13] and [12] namely WOAL and SCAL. We can remark that the proposed algorithm in this paper has much better performance in general. The interpretation for such enhanced performance is that in the WOAL Lèvy walk was applied after whale steps and hence such cascaded walk style without updating the global best disturbs the walk and the optimizer with such cascaded walk can miss the optimum. Moreover, in SCAL the Lèvy is adopted only if stagnation occurs for a search agent and hence it is rarely applied while Lèvy has its own balance between exploration and exploitation and can be used with much higher rate safely.

Contrasting the proposed LWOA to CS which also adopts Lèvy flight we can remark that both optimizers have comparable performance at the composite test functions while for the rest of function we can see that LWOA has better performance as it still depends on the exploitative capabilities of WOA which have good performance at fitness function with less complexity. A similar remark and conclusion can be deduced for LSCA.

In comparison to PSO, we can see that the appropriate balance between exploration and exploitation the proposed LWOA and LSCA has better performance than PSO especially for composite and multi-modal test functions where PSO relies on random number generators that use uniform distribution with a finite variance which makes the walk steps much limited and hence liable to stagnation.

Similar conclusions can be derived as above but with much less enhance can be seen from the data in Tables 9 and 10 for outliers values for β. such remark recommends an appropriate setting for β so that we can get good performance adopting Lèvy flights.

Tables 11–14 outline the standard deviation of performance for the different optimizers adopted in the study. Standard deviation is used to test the optimizer's capability to converge to same/similar optima at the different runs of the optimizer. We can see from the tables that the adoption of Lèvy flights to the optimization did not add to the standard deviation of performance although it uses random steps with infinite variance. Such remark can be interpreted by the fact that the exploitative mechanisms used help limiting the steps from going to infinity and also Lèvy generates a small fraction of steps of such high magnitude and hence we can say that even with Lèvy flights both LWOA and LSCA can provide repeatable robust results.

Tables 15 and 16 outline two significance measures over the proposed variants against the other optimizers. As can be seen from the tables, the significant enhancement in performance of LWOA and LSCA is much related to the setting of the β parameter where we can see significant enhance over PSO, WOA, and

Table 17
Run time average for the different optimizers adopted in the study for the different functions.

Func.	LWOA	WOAL	WOA	CS	PSO	SCA	SCAL	LSCA
F01	0.579	1.048	**0.136**	0.164	1.494	0.291	0.326	0.582
F02	0.596	1.308	0.111	0.191	**0.064**	0.163	0.227	0.463
F03	0.978	1.540	0.489	0.565	**0.444**	0.553	0.600	0.890
F04	0.579	1.078	0.098	0.185	**0.048**	0.148	0.145	0.436
F05	0.613	1.120	0.117	0.263	**0.067**	0.160	0.207	0.481
F06	0.598	1.047	0.095	0.168	**0.050**	0.150	0.231	0.483
F07	0.709	1.185	0.162	0.240	**0.119**	0.211	0.338	0.492
F08	0.649	1.144	0.111	0.205	**0.077**	0.163	0.249	0.473
F09	0.581	1.084	0.092	0.210	**0.060**	0.147	0.225	0.460
F10	0.575	1.110	0.107	0.200	**0.075**	0.174	0.207	0.462
F11	0.619	1.089	0.123	0.198	**0.086**	0.171	0.242	0.462
F12	0.765	1.317	0.306	0.377	**0.258**	0.323	0.413	0.670
F13	0.776	1.358	0.329	0.398	**0.276**	0.325	0.440	0.659
F14	0.766	0.824	0.804	0.847	**0.699**	0.778	0.856	1.377
F15	0.126	0.204	0.069	0.132	**0.046**	0.069	0.106	0.308
F16	0.078	0.116	0.053	0.125	**0.029**	0.050	0.089	0.277
F17	0.072	0.110	0.043	0.136	**0.023**	0.044	0.108	0.265
F18	0.072	0.115	0.045	0.107	**0.023**	0.047	0.086	0.263
F19	0.147	0.205	0.104	0.180	**0.081**	0.107	0.166	0.341
F20	0.195	0.315	0.137	0.175	**0.084**	0.118	0.170	0.341
F21	0.219	0.316	0.184	0.251	**0.134**	0.161	0.245	0.405
F22	0.258	0.357	0.229	0.315	**0.184**	0.208	0.316	0.451
F23	0.321	0.424	0.309	0.403	**0.256**	0.274	0.362	0.510
F24	34.721	35.267	33.880	**33.716**	34.404	35.216	34.674	35.355
F25	**33.824**	33.935	33.980	34.792	34.295	35.627	35.194	35.077
F26	35.929	33.790	**33.500**	34.217	34.155	34.409	35.297	35.141
F27	39.412	**37.797**	38.900	38.010	39.590	39.462	39.350	38.214
F28	40.879	**37.716**	37.825	38.103	39.832	38.929	39.854	38.326
F29	**35.890**	37.851	37.454	38.924	38.909	39.092	39.431	39.113

WOAL. The significance is less apparent in comparing LWOA to CS where it also adopts Lèvy flight and uses very clever exploration mechanism. At the outlier values for β, the significance almost disappear were random differences can be seen. The same behavior is apparent also in LSCA contrasting it to SCA, PSO, and SCAL.

A final Table 17 outlines the runtime in millisecond per optimizer and per function. As apparent from the table, we can see that PSO has the minimum run time for its simple and direct repositioning mechanisms and it relies on the simplest random number generator which is uniform random number generator (RNG). We can see also that WOAL is the worst optimizer with respect to run time as it consumes time applying WOA walk then consumes time to apply Lèvy walk and hence its runtime is almost doubled. SCAL also consumes more run time for stagnation detection process which interprets its higher run time. On the other hand, we can see that the extra time consumed for adding Lèvy to WOA or SCA is a very minor as the algorithm switches between the two walks so the extra time consumed is just the runtime difference between RNG based on uniform distribution and based on Lèvy distribution which is very minor.

5. Conclusion and future work

In this paper, a Lèvy flight-based variant of two recently proposed optimizers was proposed and evaluated on a set of test functions. The proposed variant was based on Lèvy flights to enhance both the exploration and exploitation capabilities of WSA and SCA. The proposed modified algorithms were compared against the original optimizers using a set of the unimodal, multimodal, and fixed-dimension multimodal as well as composite benchmark functions using a set of assessment indicators. Results prove the capability of the proposed Lèvy flight based variant to outperform the basic algorithms thanks to its much explorative capability and such capability is much apparent in the case of complex search spaces with many local minima. Furthermore, the proposed variants reduce optimizer's stagnation probability and premature convergence. Moreover, the results found out that the proposed variants can have repeatable results by converging to same/similar optima regardless of the initial search agents' locations. Moreover, the enhanced performance of the proposed variants was ensured to be significant. On the basis of future performance, we have three ideas that can be investigated in addition to the work presented here:

1. The proposed LWSA and LSCA methods will be assessed using complex datasets that have a huge number (thousands) of input features.
2. Add more statistics evaluation measures such as (sensitivity, specificity, and F-measure).
3. Apply LWSA and LSCA methods to solve challenging problems in different applications like big data, bioinformatics, and biomedical.

Acknowledgment

This work was partially supported by the IPROCOM Marie Curie initial training network, Romania, funded through the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme FP7/2007–2013/ under REA grant agreement No. 316555. This fund only applies to one co-author (Hossam M. Zawbaa).

References

[1] C.L. Huang, C.J. Wang, A GA-based feature selection and parameters optimization for support vector machines, Expert Syst. Appl. 31 (2) (2006) 231–240.
[2] C.L. Huang, J.F. Dun, A distributed PSO-SVM hybrid system with feature selection and parameter optimization, Appl. Soft Comput. 8 (4) (2008) 1381–1391.
[3] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, Global Optim. 39 (3) (2007) 459–471.
[4] X.L. Li, Z.J. Shao, J.X. Qian, An optimizing method based on autonomous animates: Fish-swarm algorithm, Methods Pract. Syst. Eng. (2002) 32–38.
[5] X.S. Yang, S. Deb, Cuckoo search via Lèvy fligh, in: Proceeding of World Congress on Nature and Biologically Inspired Computing (NaBIC 2009), India, IEEE Publications, USA, 2009, pp. 210–214.
[6] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, Knowl.-Based Syst. 89 (2015) 228–249.

[7] I. Pavlyukevich, Lèvy flights non-local search and simulated annealing, Comput. Phys. 226 (2007) 1830–1844.
[8] G.M. Viswanathan, Lévy flights and superdiffusion in the context of biological encounters and random searches, Phys. Life Rev. 5 (2008) 133–150.
[9] P. Barthelemy, J. Bertolotti, D.S. Wiersma, A Lèvy flight for light, Nature 453 (2008) 495–498.
[10] A.M. Reynolds, M.A. Frye, Free-flight odor tracking in Drosophila is consistent with an optimal intermittent scale-free search, PLoS One 2 (4) (2007) e354.
[11] C. Brown, L.S. Liebovitch, R. Glendon, Lèvy flights in Dobe Ju'hoansi foraging patterns, Hum. Ecol. 35 (2007) 129–138.
[12] L. Ning, L. Gang, D. ZhongLiang, An improved sine cosine algorithm based on Levy flight, in: Ninth International Conference on Digital Image Processing (ICDIP), Hong Kong, China, 2017.
[13] Y. Ling, Y. Zhou, Q. Luo, Levy flight trajectory-based whale optimization algorithm for global optimization, IEEE Access 5 (2017) 6168–6186.
[14] E. Emary, Hossam M. Zawbaa, Feature selection via Lèvy Antlion optimization, Pattern Anal. Appl. (PAAA) (2018).
[15] S. Mirjalili, SCA: A sine cosine algorithm for solving optimization problems, Knowl.-Based Syst. 96 (2016) 120–133.
[16] Ahmed Hafez, Hossam M. Zawbaa, E. Emary, Aboul Ella Hassanien, Sine cosine optimization algorithm for feature selection, in: International Symposium on INnovations in Intelligent SysTems and Applications (INISTA), Romania, 2016, pp. 1–5.
[17] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.
[18] E. Emary, Hossam M. Zawbaa, Mustafa Abdul Salam, A proposed whale search algorithm with adaptive random walk, in: IEEE 13th International Conference on Intelligent Computer Communication and Processing (ICCP), Romania, 2017.
[19] X.S. Yang, Nature-Inspired, Metaheuristic Algorithms, Luniver Press, 2010.
[20] X.S. Yang, Z. Cui, R. Xiao, A.H. Gandomi, M. Karamanoglu, Swarm Intelligence and Bio-Inspired Computation, Elsevier, London, 2013, pp. 18–20.
[21] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, IEEE Trans. Evol. Comput. 3 (1999) 82–102.
[22] J. Digalakis, K. Margaritis, On benchmarking functions for genetic algorithms, Int. J. Comput. Math. 77 (2001) 481–506.
[23] M. Molga, C. Smutnicki, Test functions for optimization needs, 2005. http://www.robertmarks.org/Classes/ENGR5358/Papers/functions.pdf.
[24] X.S. Yang, Firefly algorithm stochastic test functions and design optimisation, Int. J. Bio-Inspired Comput. 2 (2) (2010) 78–84.
[25] R. Salomon, Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. A survey of some theoretical and practical aspects of genetic algorithms, BioSystems 39 (1996) 263–278.
[26] J. Liang, P. Suganthan, K. Deb, Novel composition test functions for numerical global optimization, in: Proceedings of the 2005 Swarm Intelligence Symposium, SIS, 2005, pp. 68–75.
[27] P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, Y. Chen, A. Auger, et al., Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization, KanGAL Report, 2005.
[28] S. Mirjalili, A. Lewis, S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization, Swarm Evol. Comput. 9 (2013) 1–14.
[29] S. Mirjalili, S.M. Mirjalili, XS. Yang, Binary bat algorithm, Neural Comput. Appl. 25 (3–4) (2014) 663–681.
[30] E. Emary, H.M. Zawbaa, A.E. Hassanien, Binary grey wolf optimization approaches for feature selection, Neurocomputing 172 (8) (2016) 371–381.
[31] E. Emary, H.M. Zawbaa, A.E. Hassanien, Binary ant lion approaches for feature selection, Neurocomputing 213 (2016) 54–65.
[32] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995, pp. 39–43.
[33] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.
[34] X.S. Yang, Firefly algorithm Lèvy flights and global optimization, in: M. Bramer, R. Ellis, M. Petridis (Eds.), Research and Development in Intelligent Systems XXVI, Springer, London, 2010, pp. 209–218.
[35] S.L. Tilahun, H.C. Ong, Prey-predator algorithm: A new metaheuristic algorithm for optimization problems, Inf. Technol. Decis. Mak. 14 (6) (2015) 1331–1352.
[36] L. Jia, W. Gong, H. Wu, An improved self-adaptive control parameter of differential evolution for global optimization, Comput. Intell. Intell. Syst. 51 (2009) 215–224.
[37] F. Wilcoxon, Individual comparisons by ranking methods, Biom. Bull. 1 (6) (1945) 80–83.
[38] J.A. Rice, Mathematical Statistics and Data Analysis, third ed, Duxbury Advanced, 2006.