IgMin Research STEM ISSN: 2995-8067 A Multidisciplinary Open Access Journal

TECHNOLOGY SUBJECT

Data Science | Machine Learning | Data Mining TOPIC(S)

Article Information

Submitted: April 04, 2024

Approved: April 17, 2024

Published: April 18, 2024

How to cite this article: Shikoun NH, Fathi IS. Improved Energy Valley Optimizer with Levy Flight for Optimization Problems. IgMin Res. April 18, 2024; 2(4): 245-254. IgMin ID: igmin172; DOI: 10.61927/igmin172; Available at: igmin.link/p172

ORCiD: https://orcid.org/0000-0002-8207-4702

Keywords: Optimization algorithms; Energy valley optimizer; Metaheuristic algorithms; Levy flight

Research Article

Improved Energy Valley Optimizer with Levy Flight for Optimization Problems

Nabila H Shikoun[1,2] and Islam S Fathi[1]*

[1]Department of Information Systems, Al Alson Higher Institute, Cairo 11762, Egypt

[2]Department of Systems and Computer Engineering, Faculty of Engineering, Alazhar University, Cairo, Egypt

***Correspondence:** Islam S Fathi, Department of Information Systems, Al Alson Higher Institute, Cairo 11762, Egypt, Email: i_said76@yahoo.com

Abstract

Energy Valley Optimizer (EVO) is one of the recent metaheuristic algorithms. It draws inspiration from advanced principles in physics related to particle stability and decay modes. This paper presents a new Energy Valley Optimizer (EVO) and levy flights that are hybrid to improve the EVO in solving optimization problems. Levy flight is one of the most important randomization techniques. Fifteen mathematical test functions (five unimodal functions, four multimodal functions, and six composite functions) are solved with the proposed algorithm. We also compare our results with previous results of metaheuristic algorithms. The statistical results show that the results of the Levy Energy Valley Optimizer (LEVO) outperform other algorithms in almost all mathematical test functions.

Introduction

Optimization algorithms may be divided into two categories: deterministic and stochastic algorithms. Deterministic algorithms don’t contain stochastic operators; they give the same answer if the initial start point is constant. On the contrary, stochastic algorithms give different answers even if the initial start point is constant [1]. Metaheuristic algorithms are the most popular techniques for solving optimization problems. Natural phenomena inspire it. Such algorithms improved local optima solutions to reach global optima in search space [2]. The interested metaheuristic algorithms in the literature are Genetic Algorithms (GA) [3], Ant Colony Optimization (ACO) [4], Particle Swarm Optimization(PSO) [5], Artificial Bee Colony (ABC) Algorithm[6], Cuckoo Optimization Algorithm (COA) [7], Firefly Algorithm (FA) [8], Grey Wolf Optimizer (GWO) [9], Equilibrium optimizer (EO) [10], Horse herd optimization algorithm (HOA) [11], drawer algorithm (DA) [12], Walrus Optimization Algorithm (WaOA) [13], and Energy Valley Optimizer (EVO). The EVO is considered one of the latest nature-inspired algorithms proposed in metaheuristic algorithms [1]. The Energy Valley Optimizer (EVO) [1] is a recently developed metaheuristic algorithm that draws inspiration from sophisticated ideas in physics pertaining to particle stability and decay modes. The idea of the EVO comes from the fundamental laws of how particles decay through different types of matter in physics. The Energy Valley Optimizer also includes an examination of the complexity of the test functions used and achieves excellent results. As mentioned before, metaheuristics have shown a positive influence on feature selection problems in recent years [14]. There is still a need for additional optimization strategies to achieve further improved results. Exploration means finding promising solutions by seeking various unknown regions, while exploitation improves over solutions obtained by exploration [15].

Levy flight is a class of random walks whose step lengths are not constant but are drawn from levy distribution, proposed by Paul Levy [16]. The foraging movements are observed to follow Levy distribution. Levy flight makes a large jump in random walks, and this allows the individual to visit new sites that the swarm has not visited, which leads to high exploration in search space [17,18].

So, this paper presents a new Energy Valley Optimizer (EVO) and levy flights that are hybrid to improve the EVO in solving optimization problems. The remainder of the paper is organized as follows: in section 2, the basic EVO is described in detail. The definition of Levy flight and Levy Energy Valley optimizer (LEVO) algorithm is presented in section 3. Section 4 gives the experimental results. Finally, section 5 provides the conclusions of this work.

Overview of the basic EVO

Main inspiration: The Energy Valley Optimizer (EVO) [1] is a new metaheuristic algorithm that has been utilized in engineering optimization issues. It is classified within the category of physics-based approaches. EVO is inspired by the fundamental laws of how particles decay through different types of matter in physics. The term "physical reaction" pertains to the process of inducing the collision of two particles or foreign subatomic particles, resulting in the formation of novel particles. It is thought that most particles are unstable, but some are stable and stay that way forever. The unstable particles give off energy when they break apart, which is also known as decay. The total decay rate is different for each type of particle. During the decay process, a particle undergoes a reduction in energy, with the surplus energy being emitted. The Energy Valley concept involves scrutinizing particle stability by analyzing binding energy and how the particles interact with each other. The central focus in this domain centers on determining particle stability, which is contingent on evaluating neutron (N) and proton (Z) quantities, as well as the N/Z ratio. An N/Z ratio approximately equal to 1 signifies the particle is stable and light, whereas a higher N/Z value indicates stability in a heavier particle. Particles tend to enhance their stability by adjusting their neutron-to-proton (N/Z) ratio, gravitating towards the region of stability or the energy minimum, as illustrated in Figure 1A.

During the decay process, the emission of excessive energy results in the generation of a particle in a reduced energy state. The decay mechanism in particles exhibiting varying degrees of stability is determined by three distinct types of emissions. The alpha (α) particles are highly dense particles with a positive charge. Beta (β) particles are electrically charged particles that possess a negative charge. These particles can be described as electrons that exhibit greater velocities. Gamma (γ) rays are photons characterized by elevated energy levels, as depicted in Figure 1B. Based on the information provided regarding the emission process, it can be observed that there exist three distinct forms of decay, namely alpha, beta, and gamma decay, which are generated from the aforementioned emission kinds. Alpha decay is characterized by the emission of an alpha particle, leading to a decrease in both the neutron (N) and proton (Z) values and consequently reducing the N/Z ratio. Conversely, beta decay involves the emission of a β particle, posing a challenge to the N/Z ratio by reducing the number of neutrons (N) and increasing the number of protons (Z).In the process of gamma decay, the emission of a gamma (γ) photon from an excited particle is observed without any accompanying alteration in the N/Z values, as shown in Figure 1C. The EVO depends on the idea that different particles decay over time. The method uses the particles' tendency to reach a stable point as a starting point for improving the performance of the candidate solutions.

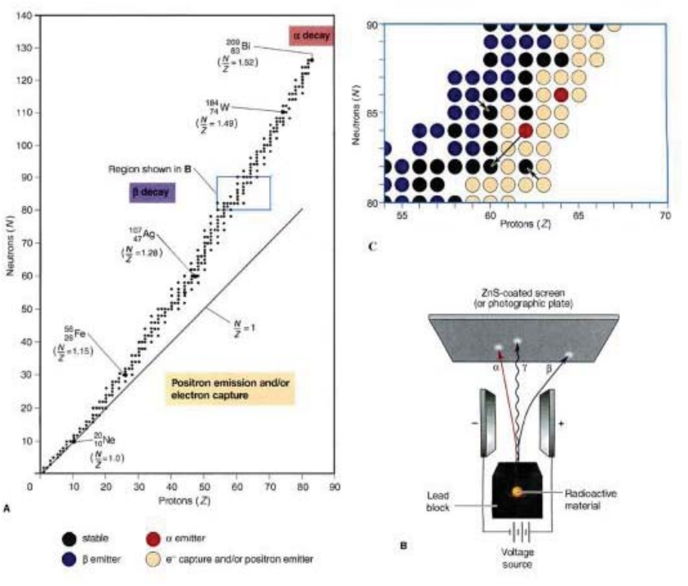


Figure 1: (A) particles stability band (B) Emission process (C) Three types of decay [1].

Mathematical formula: The first phase involves the execution of the initialization procedure, during which the solution candidates (P_i) are conceptualized as particles that have varying degrees of stability inside the search space.

$$P = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_i \\ \vdots \\ P_n \end{bmatrix} = \begin{bmatrix} p_1^1 p_1^2 \cdots p_1^j \cdots p_1^d \\ p_2^1 p_2^2 \cdots p_2^j \cdots p_2^d \\ \cdots\cdot\ddots \\ \cdots \\ p_i^1 p_i^2 \cdots p_i^j \cdots p_i^d \\ \cdots\cdot\vdots \\ p_n^1 p_n^2 \cdots p_n^j \cdots p_n^d \end{bmatrix}, \begin{cases} i = 1, 2, \ldots, n. \\ j = 1, 2, \ldots, d. \end{cases} \quad (1)$$

$$p_i^j = p_{i,min}^j + \text{rand} \cdot \left(p_{i,max}^j - p_{i,min}^j \right), \begin{cases} i = 1, 2, \ldots, n. \\ j = 1, 2, \ldots, d. \end{cases} \quad (2)$$

The variable "n" represents the particle number within the search space. The variable "d" represents the problem dimension under consideration. p_i^j represents the j-th decision variable used to estimate the initial position of the i-th candidate. $p_{i,max}^j$ and $p_{i,min}^j$ are the upper and lower bounds of the j-th variable in the i-th candidate. The variable "rand" represents a random number that follows a uniform distribution inside the interval [0,1].

The second phase of the method involves determining the Enrichment Bound (EB) for the particles. This parameter is employed to accommodate variations between particles with a surplus of neutrons and particles with a deficit of neutrons.

In order to accomplish this, the evaluation function is assessed for every particle, leading to the determination of the Neutron Enrichment Level (NEL) of these particles. The aforementioned elements are mathematically represented in the following manner:

$$\mathrm{EB} = \frac{\sum_{i=1}^{n} \mathrm{NEL}_i}{\mathrm{n}}, \quad \mathrm{i} = 1, 2, \ldots, \mathrm{n}. \tag{3}$$

The variable NEL_i represents a level of neutron enrichment for the i-th particle, while EB denotes particles that are the enrichment bound.

In the third phase, the estimation of particle stability values is conducted by evaluating the objective function.

$$\mathrm{SL}_i = \frac{\mathrm{NEL}_i - \mathrm{BS}}{\mathrm{WS} - \mathrm{BS}}, \quad \mathrm{i} = 1, 2, \ldots, \mathrm{n}. \tag{4}$$

The stability value for the i-th particle is SL_i, is determined based on the best (BS) and worst (WS) stability levels. These levels correspond to the lowest and highest objective function values discovered thus far. In the EVO's main search loop, if a particle's neutron enrichment level NEL_i surpasses the enrichment limit (EB), which implies that the particle has a higher neutron-to-proton (N/Z) ratio. Furthermore, alpha and gamma decays are anticipated to occur if the particle›s stability level SL_i exceeds the stability bound (SB). This expectation stems from the likelihood of such decay in larger particles with elevated stability levels. In the physics of alpha decay (as illustrated in Figure 2). The decision variables present in the solution candidate are substituted by the rays within the particle or candidate exhibiting the highest level of stability, referred to as P_{BS}. The mathematical representation of these is as follows:

$$\mathrm{P}_i^{New1} = \mathrm{P}_i\left(\mathrm{P}_{\mathrm{BS}}\left(\mathrm{p}_i^j\right)\right), \begin{cases} i = 1, 2, \ldots, n. \\ j = Alpha\ Index\ II. \end{cases} \tag{5}$$

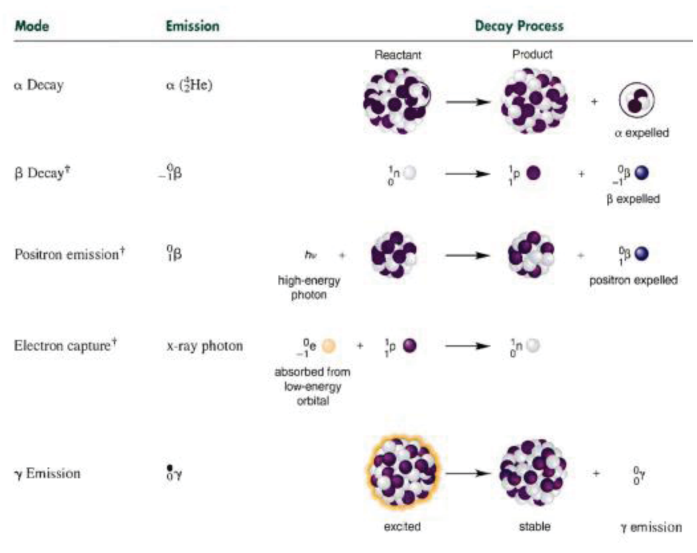


Figure 2: Different forms of decay [1].

Where P_i^{New1} represents a newly produced particle within the search space, P_i is the current position vector of an i-th particle in search space. P_{BS} is the particle's position vector that has the highest level of stability, p_i^j is the j-th decision variable.

In this context, the estimation of the total distance between the particle under consideration and other particles is performed using a method, and the nearest particle is selected for this purpose.

$$\mathrm{D}_i^k = \sqrt{\left(\mathrm{x}_2 - \mathrm{x}_1\right)^2 + \left(\mathrm{y}_2 - \mathrm{y}_1\right)^2}, \begin{cases} i = 1, 2, \ldots, n. \\ k = 1, 2, \ldots, n-1. \end{cases} \tag{6}$$

The variable D_i^k represents the overall distance separating the i-th particle from the k-th neighboring particle. The coordinates of the particles are denoted by (x1, y1) and (x2, y2).

The procedure of updating the position to generate the second solution candidate in this phase is carried out by employing the following actions:

$$\mathrm{P}_i^{New2} = \mathrm{P}_i\left(\mathrm{P}_{\mathrm{Ng}}\left(\mathrm{p}_i^j\right)\right), \begin{cases} i = 1, 2, \ldots, n. \\ j = Gamma\ Index\ II. \end{cases} \tag{7}$$

P_i^{New2} is a newly generated particle

P_i is the current position vector of the i-th particle

P_{Ng} is the position vector of the neighboring particle around the i-th particle

p_i^j is the j-th decision variable or emitted ray.

Beta decay takes place in particles with lower stability levels, indicating less stability. According to the principles of physics for beta decay, illustrated in Figure 2, particles release β rays to enhance their stability. Given the instability of these particles, a significant leap in the search space is necessary. In such cases, a procedure is employed to update the particle positions, involving controlled movements toward the particle or option with the best stability level (P_{BS}) and the center of the particles (P_{CP}). This aspect of the algorithm mimics the particles' inclination to approach the stability band. Particles are situated near this band, and the majority of them exhibit higher stability (refer to Figure 1a and Figure 1b). These concepts are expressed as follows:

$$\mathrm{P}_{\mathrm{CP}} = \frac{\sum_{i=1}^{n} \mathrm{P}_i}{\mathrm{n}}, i = 1, 2, \ldots, n. \tag{8}$$

$$\mathrm{P}_i^{\mathrm{New1}} = \mathrm{P}_i + \frac{\left(\mathrm{r}_1 \times \mathrm{P}_{\mathrm{BS}} - \mathrm{r}_2 \times \mathrm{P}_{\mathrm{CP}}\right)}{\mathrm{SL}_i}, \quad \mathrm{i} = 1, 2, \ldots, \mathrm{n}. \tag{9}$$

$\mathrm{P}_i^{\mathrm{New1}}$ the upcoming position vectors of i-th particles.

P_i the current position vectors of i-th particles.

P_{BS} is the particle position vector of the optimal stability level.

P_{CP} is the centre of the particle position vector

SL_i is the level of stability for the i-th particle.

The parameters r_1 and r_2 represent two randomly generated integers within the range of [0, 1].

To make the algorithm better at exploitation and exploration, a different process is used to update the positions of particles that use beta decay. This procedure entails guiding the particles systematically toward the particle with the optimal stability level (P_{BS}) and a neighboring particle or candidate (P_{Ng}), ensuring that the movement is not influenced by particle stability level. These elements can be mathematically articulated as follows:

$$P_i^{New2} = P_i + \left(r_3 \times P_{BS} - r_4 \times P_{Ng}\right), \quad i = 1, 2, \ldots, n. \tag{10}$$

P_i^{New2} is the forthcoming position vectors of i-th particles.

P_i is i-th particles' current position vectors.

P_{BS} is the particle position vector of the optimal stability value.

P_{Ng} is the neighbouring particle's position vector around the i-th particle.

The parameters r_3 and r_4 represent two randomly generated integers within the range of [0, 1].

When the neutron enrichment level (NEL_i) of a particle below the enrichment bound (EB), it is postulated that the particle possesses a reduced neutron-to-proton (N/Z) ratio. To approach the stability band, the particle has a tendency to either absorb electrons or emit positrons. In this context, a stochastic movement within the search space is characterized to accommodate these types of motions, expressed as follows:

$$P_i^{New} = P_i + r\ , \quad i = 1, 2, \ldots, n. \tag{11}$$

where P_i^{New} and P_i are the forthcoming and current position vectors of i-th particles.

The parameter r represents a randomly generated integer within the range of [0, 1].

Description Energy Valley Optimizer (LEVO)

Concept of levy flight: Levy flight is a kind of random walk whose step length is not constant, but it is drawn from levy distribution. Levy distribution has infinite variance and infinite mean with power low step size [19,20]. Some animals and insects, such as ants, are following levy flight in walk-in foraging patterns [21]. Levy distribution is useful for stochastic algorithms. It has a role in exploration and exploitation [22]. Levy flight is expressed mathematically as:

$$levywalk = 0.01 \times \frac{x}{\left|y\right|^{\frac{1}{\beta}}} \tag{12}$$

Where x and y are random numbers drawn from a normal distribution.

$$x \sim N(0, \sigma_x^2),\ y \sim N(0, \sigma_y^2) \tag{13}$$

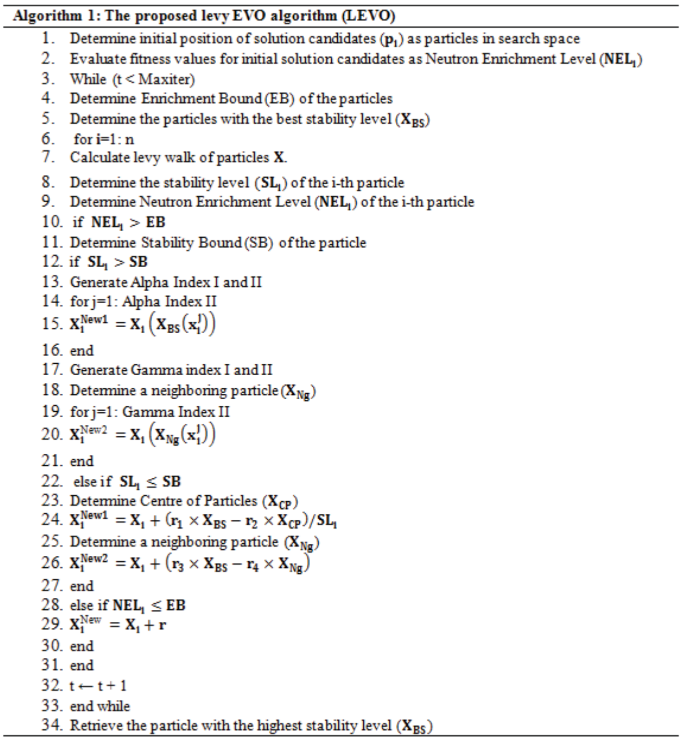
Where

$$\sigma_x = \left(\frac{\Gamma(1+\beta) \times \sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{(\frac{\beta-1}{2})}} \right)^{1/\beta}, \sigma_y = 1,\ \Gamma\left(n\right) = \left(n-1\right)! \text{ and } 0 < \beta \leq 2, \beta = 3/2$$

The proposed levy EVO algorithm (LEVO): The proposed (LEVO) is the improved form of the original EVO, where hybrid EVO with levy flight. In Algorithm 1, the proposed algorithm is represented in simple form.

Experimental results and discussions

The results of fifteen benchmark functions are presented



```
Algorithm 1: The proposed levy EVO algorithm (LEVO)
Determine initial position of solution candidates ($\mathbf{p}_1$) as particles in search space
Evaluate fitness values for initial solution candidates as Neutron Enrichment Level ($\mathbf{NEL}_1$)
While (t < Maxiter)
    Determine Enrichment Bound (EB) of the particles
    Determine the particles with the best stability level ($\mathbf{X}_{BS}$)
    for i=1: n
        Calculate levy walk of particles X.
        Determine the stability level ($\mathbf{SL}_1$) of the i-th particle
        Determine Neutron Enrichment Level ($\mathbf{NEL}_1$) of the i-th particle
        if $\mathbf{NEL}_1 > \mathbf{EB}$
            Determine Stability Bound (SB) of the particle
            if $\mathbf{SL}_1 > \mathbf{SB}$
                Generate Alpha Index I and II
                for j=1: Alpha Index II
                    $\mathbf{X}_1^{New1} = \mathbf{X}_1\left(\mathbf{X}_{BS}(\mathbf{x}_1^j)\right)$
                end
                Generate Gamma index I and II
                Determine a neighboring particle ($\mathbf{X}_{Ng}$)
                for j=1: Gamma Index II
                    $\mathbf{X}_1^{New2} = \mathbf{X}_1\left(\mathbf{X}_{Ng}(\mathbf{x}_1^j)\right)$
                end
            else if $\mathbf{SL}_1 \leq \mathbf{SB}$
                Determine Centre of Particles ($\mathbf{X}_{CP}$)
                $\mathbf{X}_1^{New1} = \mathbf{X}_1 + (\mathbf{r}_1 \times \mathbf{X}_{BS} - \mathbf{r}_2 \times \mathbf{X}_{CP})/\mathbf{SL}_1$
                Determine a neighboring particle ($\mathbf{X}_{Ng}$)
                $\mathbf{X}_1^{New2} = \mathbf{X}_1 + (\mathbf{r}_3 \times \mathbf{X}_{BS} - \mathbf{r}_4 \times \mathbf{X}_{Ng})$
            end
        else if $\mathbf{NEL}_1 \leq \mathbf{EB}$
            $\mathbf{X}_1^{New} = \mathbf{X}_1 + \mathbf{r}$
        end
    end
    $t \leftarrow t + 1$
end while
Retrieve the particle with the highest stability level ($\mathbf{X}_{BS}$)
```

Algorithm 1

in this section. These test functions can be classified into three categories. The first category, unimodal test functions, contains five test functions. The second contains four functions of multimodal functions. The other six problems are composite test functions located in the last category. Tables 1-3 present the test function survey of each category. The proposed LEVO is compared with the basic Energy valley optimizer (EVO) [1] and recent algorithms such as Genetic Algorithms (GA) [3], Particle Swarm Optimization (PSO) [5], Firefly Algorithm (FA) [8], states of matter search (SMS) [23], Bat algorithm (BA) [24], Flower pollination algorithm (FPA) [25], and Cuckoo search (CS) [26]. The proposed LEVO algorithm uses 30 candidate solutions (particles) over 1000 iterations. For each category, two tables are given. The first table includes the values of the decision variable and the objective function of the best run out of 30 independent runs. The second table contains Ave and std. Ave is the mean of the optimal objective value calculated for 30 independent runs. Std is the standard deviation of optimal objective values Table 4.

Results on unimodal benchmark functions

The Unimodal function is the function that has a single optimum. Test functions are presented in Table 1. Table 5 presents the best values of objective functions among 30 independent runs and their corresponding decision variables (x's). Statistical results are listed in Table 6. Results show that, on most test functions, the proposed LEVO algorithm gets better results than other algorithms.

Table 1: Unimodal test functions.

Test function	domain	Dim	F_{opt}
$F_1(x) = \Sigma_{i=1}^{n} x_i^2$	[-100,100]	30,200	0
$F_2(x) = \Sigma_{i=1}^{n} \lvert x_i \rvert + \Pi_{i=1}^{n} \lvert x_i \rvert$	[-10,10]	30,200	0
$F_3(x) = \Sigma_{i=1}^{n} \left(\Sigma_{j=1}^{i} x_j\right)^2$	[-100,100]	30,200	0
$F_4(x) = \max_i \{\lvert x_i \rvert, 1 \le i \le n\}$	[-100,100]	30,200	0
$F_5(x) = \Sigma_{i=1}^{n} i x_i^4 + random[0,1)$	[-1.28, 1.28]	30,200	0

Table 2: Multimodal test functions.

Test function	domain	Dim	F_{opt}
$F_6(x) = \Sigma_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$	[-500,500]	30,200	-418.9829×Dim^a
$F_7(x) = \Sigma_{i=1}^{n} x_i^2 - 10\cos(2\pi x_i) + 10]$	[-5.12, 5.12]	30,200	0
$F_8(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\Sigma_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\Sigma_{n=1}^{i} \cos(2\pi x_i)) + 20 + e$	[-32, 32]	30,200	0
$F_9(x) = \frac{1}{4000}\Sigma_{n=1}^{i} x_i^2 - \Pi_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600,600]	30,200	0

Table 3: Composite test functions (mathematical formulation of the sphere, Ackley, Griewank, Weierstrass, Rastrigin illustrated in Table 4).

Test function	Domain	Dim	F_{OPT}
F_{10}(CF1):	[-5, 5]	10	0
$f_1,f_2,f_3,……f_{10}$= Sphere Function			
[$\sigma_1,\sigma_2,\sigma_3,… \sigma_{10}$] = [1,1,1,….,1]			
[$\lambda_1,\lambda_2,\lambda_3,….,\lambda_{10}$=[5/100, 5/100, 5/100,…,5/100]			
F_{11}(CF2):	[-5, 5]	10	0
$f_1,f_2,f_3,……f_{10}$=Griewank's Function			
[$\sigma_1,\sigma_2,\sigma_3,… \sigma_{10}$] = [1,1,1,….,1]			
[$\lambda_1,\lambda_2,\lambda_3,….,\lambda_{10}$=[5/100, 5/100, 5/100,…,5/100]			
F_{12}(CF3):	[-5, 5]	10	0
$f_1,f_2,f_3,……f_{10}$=Griewank's Function			
[$\sigma_1,\sigma_2,\sigma_3,… \sigma_{10}$]=[1,1,1,….,1]			
[$\lambda_1,\lambda_2,\lambda_3,….,\lambda_{10}$=[1,1,1,….,1]			
F_{13}(CF4):	[-5, 5]	10	0
f_1,f_2 =Ackley's Function			
f_3,f_4 =Rastrigin's Function			
F_5, f_6 =Weierstrass Function			
F_7, f_8 =Griewank'sFunction			
F_9, f_{10} =Sphere Function			
[$\sigma_1,\sigma_2,\sigma_3,… \sigma_{10}$]=[1,1,1,….,1]			
[$\lambda_1,\lambda_2,\lambda_3,….,\lambda_{10}$]=[5/32, 5/32, 1, 1, 5/0.5, 5/0.5, 5/100, 5/100, 5/100, 5/100]			
F_{14}(CF5):	[-5, 5]	10	0
f_1,f_2=Rastrigin's Function			
f_3,f_4=Weierstrass Function			
F_5, f_6=Griewank's Function			
F_7, f_8= Ackley's Function			
F_9, f_{10}= Sphere Function			
[$\sigma_1,\sigma_2,\sigma_3,… \sigma_{10}$]=[1,1,1,….,1]			
[$\lambda_1,\lambda_2,\lambda_3,….,\lambda_{10}$]=[1/5, 1/5, 5/0.5, 5/0.5, 5/100, 5/100, 5/32, 5/32, 5/100, 5/100]			
F_{15}(CF6):	[-5, 5]	10	0
f_1,f_2 =Rastrigin's Function			
f_3,f_4 =Weierstrass Function			
F_5,f_6 =Griewank'sFunction			
F_7, f_8=Ackley's Function			
F_9,f_{10}=Sphere Function			
[$\sigma_1,\sigma_2,\sigma_3,… \sigma_{10}$]=[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]			
[$\lambda_1,\lambda_2,\lambda_3,….,\lambda_{10}$]=[0.1* 1/5, 0.2 * 1/5, 0.3 * 5/0.5,0.4 * 5/0.5,0.5 * 5/100, 0.6 * 5/100, 0.7 * 5/32, 0.8 * 5/32, 0.9 * 5/100, 1 * 5/100]			

Table 4: mathematical formulation of the basic functions in Table 3.

Name	Formulation
Sphere	$f(x) = \Sigma_{i=1}^{D} x_i^2$
Ackley	$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\Sigma_{i=1}^{D} x_i^2}\right) - \exp(\frac{1}{D}\Sigma_{i=1}^{D} \cos(2\pi x_i)) + 20 + e$
Griewank	$\frac{1}{4000}\Sigma_{i=1}^{D} x_i^2 - \Pi_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}}) + 1$
Weierstrass kmax=20	$\Sigma_{i=1}^{D}(\Sigma_{k=0}^{kmax}[a^k \cos(2\pi b^k (x_i + 0.5))]) - D\Sigma_{k=0}^{kmax}[a^k \cos(2\pi b^k 0.5)], a=0.5, b=3,$
Rastrigin	$\Sigma_{i=1}^{D}(x_i^2 - 10\cos(2\pi x_i) + 10)$

Table 5: The best values of objective functions of the LEVO algorithm on unimodal functions.

	Values of the best variables				
	F1	F2	F3	F4	F5
X1	-2.10E-07	-2.10E-08	2.60E-08	4.79E-09	-1.76E-04
X2	1.07E-07	1.07E-08	-9.08E-08	-1.88E-08	1.45E-06
X3	-2.22E-09	-2.22E-10	-3.63E-07	6.45E-08	1.06E-05
X4	-2.15E-08	-2.22E-10	8.78E-09	2.93E-07	4.92E-06
X5	-1.12E-07	-1.12E-08	4.61E-07	-2.33E-07	3.42E-06
X6	5.80E-08	5.80E-09	-2.07E-07	-5.14E-08	5.23E-05
X7	7.34E-08	7.34E-09	4.49E-08	-2.78E-08	-2.48E-07
X8	4.52E-08	4.52E-09	2.86E-08	1.48E-07	3.78E-06
X9	-4.38E-08	-4.38E-09	5.15E-08	9.03E-08	3.56E-05
X10	8.16E-08	8.16E-09	-1.25E-07	6.18E-08	-8.54E-06
X11	3.56E-07	3.56E-08	1.70E-07	-2.57E-08	-5.89E-06
X12	-8.56E-08	-8.56E-09	3.17E-08	-2.08E-07	1.57E-06
X13	-1.85E-07	-1.85E-08	-3.64E-09	3.83E-07	-4.95E-06
X14	-7.16E-08	-7.16E-09	-9.02E-08	3.31E-08	-1.09E-07
X15	1.06E-08	1.06E-09	4.12E-07	-1.56E-08	1.64E-05
X16	5.90E-08	5.90E-09	-1.08E-07	1.79E-07	-1.79E-05
X17	-6.90E-08	-6.90E-09	-7.99E-07	6.98E-08	-1.14E-05
X18	1.97E-07	1.97E-08	5.87E-07	-2.95E-07	2.07E-05
X19	-5.49E-08	-5.49E-09	1.10E-07	1.44E-07	6.74E-07
X20	-3.25E-07	-3.25E-08	1.15E-08	1.87E-07	-1.05E-05
X21	1.74E-07	1.74E-08	-6.86E-08	-1.51E-07	-2.55E-05
X22	-4.02E-07	-4.02E-08	9.02E-08	1.09E-07	3.26E-06
X23	-1.60E-07	-1.60E-08	2.71E-07	-9.51E-08	-4.45E-05
X24	1.28E-08	1.28E-09	-4.58E-07	-3.25E-07	-8.32E-06
X25	3.37E-08	3.37E-09	3.89E-07	-1.89E-08	5.01E-07
X26	1.67E-07	1.67E-08	1.06E-07	1.64E-07	1.17E-05
X27	2.26E-07	2.26E-08	-8.15E-07	-4.10E-08	5.12E-06
X28	-5.98E-09	-5.98E-10	-1.30E-07	-5.48E-08	6.01E-06
X29	-7.31E-08	-7.31E-09	6.04E-07	4.26E-07	-3.76E-06
X30	3.53E-08	3.53E-09	-2.31E-07	-1.03E-07	-1.84E-05
F(x)	7.22E-13	3.46E-07	1.96E-12	4.26E-07	2.45E-07

Results on multimodal benchmark functions

The Multimodal function is a function that has more than one optimum in which there is a single global optimum. These test functions are presented in Table 2. Table 7 shows the results of the proposed LEVO on multimodal function; the average of the minimum value (ave) and their standard deviation (std) are presented in Table 8. It is clear that the LEVO algorithm has the best results in many test functions.

Results on composite benchmark functions

Composite benchmark functions are listed in Table 3. Table 9 shows the results of composite functions using LEVO. The average (Ave) and their standard deviation (std) are presented in Table 10. Results show that the proposed LEVO algorithm outperforms other algorithms on the majority of the test functions.

The convergence curves of algorithms on some of the test functions are illustrated in Figure 3.

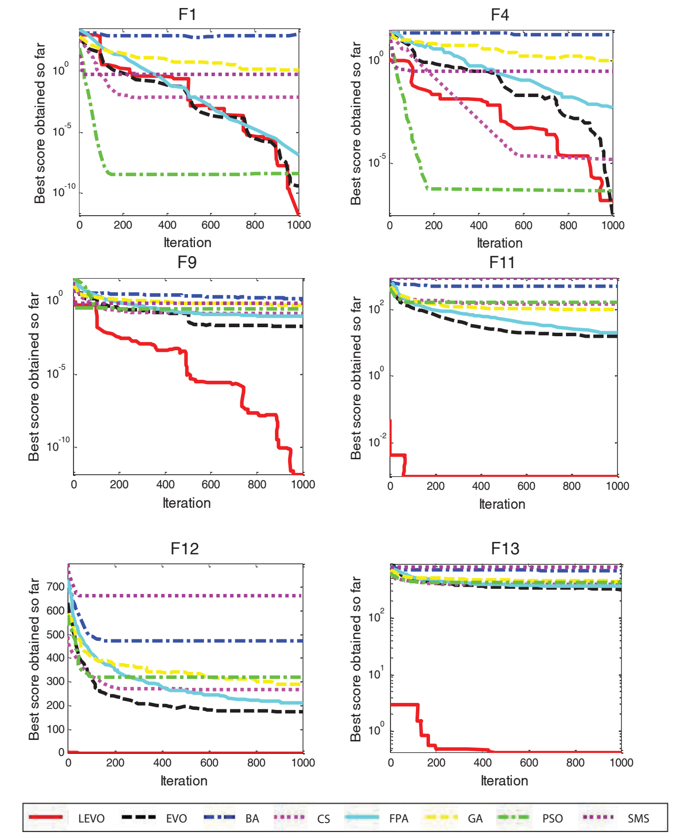


Figure 3: Convergence curves of algorithms on six of the test functions.

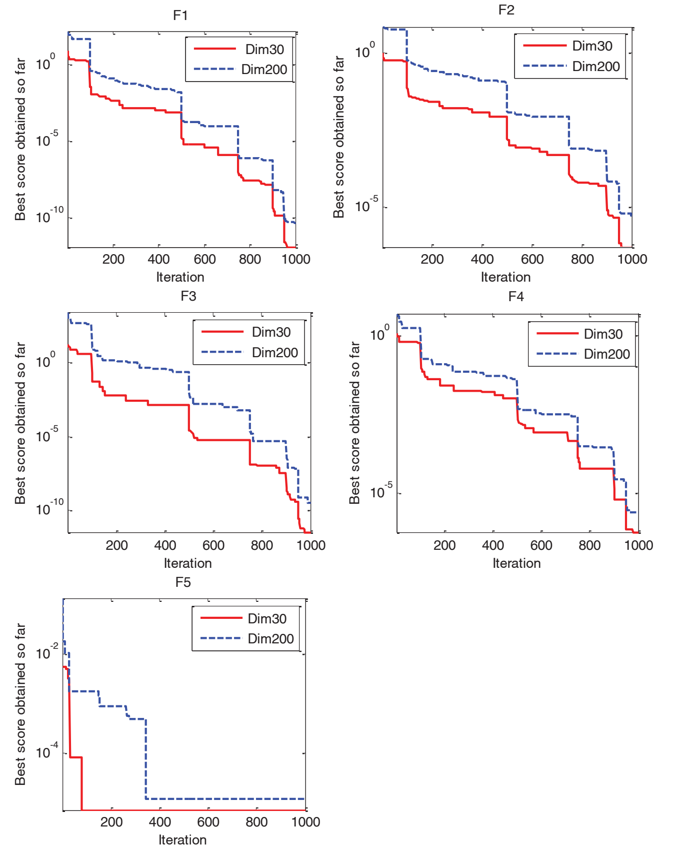


Figure 4: Behaviors of LEVO in small and large dimensions in unimodal test functions.

Table 6: Results of LEVO and other algorithms on unimodal test functions.

Function	LEVO		EVO		PSO		BA	
	Ave	std	Ave	std	Ave	Std	Ave	std
F1	1.56E-12	4.28E-13	2.59E-10	1.65E10	2.70E-09	1.00E-09	7.74E-01	5.28E-01
F2	4.67E-07	5.18E-08	1.84E-06	6.58E-07	7.15E-05	2.26E-05	3.35E-01	3.82E+00
F3	4.35E-12	1.61E-12	6.07E-10	6.34E-10	4.71E-06	1.49E-06	1.15E-01	7.66E-01
F4	5.79E-07	8.66E-08	1.36E-08	1.81E-09	3.25E-07	1.02E-08	1.92E-01	8.90E-01
F5	3.74E-05	3.67E-05	4.29E-03	5.09E-03	1.40E-03	1.27E-03	1.38E-01	1.13E-01
Function	SMS		FPA		CS		GA	
	Ave	std	Ave	Std	Ave	Std	Ave	std
F1	5.70E-02	1.47E-02	1.06E-07	1.27E-07	6.50E-03	2.05E-04	1.19E-01	1.26E-01
F2	6.85E-03	1.58E-03	6.24E-04	1.76E-04	2.12E-01	3.98E-02	1.45E-01	5.32E-02
F3	9.60E-01	8.24E-01	5.67E-08	3.90E-08	2.47E-01	2.14E-02	1.39E-01	1.21E-01
F4	2.77E-01	5.74E-03	3.84E-03	2.19E-03	1.12E-05	8.25E-06	1.58E-01	8.62E-01
F5	3.04E-04	2.58E-04	3.11E-03	1.37E-03	1.32E-03	7.28E-04	1.01E-02	3.26E-03

Table 7: The best values of objective functions of the LEVO algorithm on multimodal functions.

	Values of the best variables			
	F6	F7	F8	F9
X1	7.73E+01	-2.03E-09	4.08E-08	2.36E-08
X2	4.17E+02	-1.16E-08	-3.64E-08	7.42E-08
X3	5.00E+02	8.72E-10	-3.98E-08	3.81E-08
X4	1.99E+02	7.60E-09	-4.31E-08	-3.84E-07
X5	5.40E+01	5.06E-09	-1.70E-08	-4.86E-07
X6	-5.00E+02	-1.19E-09	2.90E-08	-1.00E-06
X7	-1.27E+02	-1.94E-10	2.84E-08	-4.42E-07
X8	-5.00E+02	-3.18E-09	2.09E-08	5.76E-07
X9	-2.38E+02	-3.66E-09	-7.83E-08	-2.52E-07
X10	1.26E+02	6.57E-09	-6.05E-08	7.32E-08
X11	2.12E+02	1.70E-08	-4.35E-08	3.38E-07
X12	2.03E+02	1.89E-09	-9.47E-08	1.15E-06
X13	4.15E+02	6.89E-09	1.37E-07	5.34E-07
X14	8.48E+01	1.94E-09	2.50E-10	-1.15E-06
X15	-2.67E+02	2.90E-09	-7.11E-08	2.11E-07
X16	-3.13E+02	1.58E-08	-4.69E-08	-5.43E-08
X17	1.86E+02	1.27E-08	1.74E-08	3.83E-07
X18	1.74E+02	-7.10E-09	-3.66E-08	3.00E-07
X19	4.56E+02	-1.61E-08	1.00E-08	3.29E-06
X20	3.61E+02	-7.21E-10	1.49E-08	9.21E-07
X21	-5.00E+02	-7.34E-09	1.25E-08	5.26E-07
X22	4.10E+02	1.32E-09	4.02E-08	1.29E-06
X23	-5.00E+02	-3.38E-09	3.22E-09	5.27E-06
X24	-3.10E+02	-4.41E-09	4.04E-08	7.72E-07
X25	3.95E+02	-4.21E-09	1.09E-08	-4.89E-08
X26	6.69E+01	1.28E-08	-5.98E-10	-1.03E-06
X27	-1.35E+01	-2.59E-09	1.21E-07	-1.46E-06
X28	-9.64E+01	1.30E-08	8.87E-09	-1.84E-07
X29	4.90E+02	1.45E-09	6.71E-09	1.11E-06
X30	1.90E+02	-1.91E-09	-1.09E-07	2.20E-07
F(x)	-4.30E+03	3.98E-13	2.16E-07	1.36E-12

Table 8: Results of LEVO and other algorithms on multimodal test functions.

Function	LEVO		EVO		PSO		BA	
	ave	std	ave	std	ave	std	ave	std
F6	-3.68E+03	2.48E+02	-1.61E+03	3.14E+02	-1.37E+03	1.46E+02	-1.07E+03	8.58E+02
F7	6.97E-13	1.88E-13	7.71E-06	8.45E-06	2.79E-01	2.19E-01	1.23E+00	6.86E-01
F8	2.81E-07	3.42E-08	3.73E-15	1.50E-15	1.11E-09	2.39E-11	1.29E-01	4.33E-02
F9	2.54E-12	6.31E-13	1.86E-02	9.55E-03	2.74E-01	2.04E-01	1.45E+00	5.70E-01
Function	SMS		FPA		CS		GA	
	ave	std	ave	std	ave	std	ave	std
F6	-4.21E+00	9.36E-16	-1.84E+03	5.04E+01	-2.09E+03	7.62E-03	-2.09E+03	2.47E+00
F7	1.33E+00	3.26E-01	2.73E-01	6.86E-02	1.27E-01	2.66E-03	6.59E-01	8.16E-01
F8	8.88E-06	8.56E-09	7.40E-03	7.10E-03	8.16E-09	1.63E-08	9.56E-01	8.08E-01
F9	7.06E-01	9.08E-01	8.50E-02	4.00E-02	1.23E-01	4.97E-02	4.88E-01	2.18E-01

Table 9: The best values of objective functions of the LEVO algorithm on composite functions.

Values of the best variables	Function					
	F10	F11	F12	F13	F14	F15
X1	-1.37E-02	1.90E-01	8.82E-02	3.13E+00	1.16E-03	5.58E-01
X2	-8.06E-04	1.29E-02	-6.64E-01	2.25E+00	-1.00E+00	5.66E-01
X3	-5.17E-04	6.73E-03	7.23E-02	8.65E-01	-1.49E-01	8.11E-01
X4	-8.79E-03	6.46E-02	-4.14E-02	3.58E+00	-6.35E-01	-9.48E-03
X5	-1.26E-02	1.99E+00	-2.06E-02	3.23E-01	-8.13E-01	2.13E-01
X6	-4.88E-03	-6.98E-02	-7.23E-04	-2.71E+00	4.75E-02	9.50E-01
X7	-1.25E-02	-1.34E-02	-4.00E-03	1.15E+00	2.82E-02	1.31E-01
X8	-9.65E-03	7.93E-03	-4.23E-02	2.93E+00	5.03E-02	1.74E+00
X9	-5.98E-03	-4.03E-02	-2.50E-02	-3.31E+00	-1.22E-02	3.96E-01
X10	-4.78E-03	3.07E-03	4.97E-03	-2.44E+00	-2.07E-01	-1.96E-01
F(x)	1.27E+01	7.15E-04	-1.01E+00	4.00E-01	3.00E+00	-1.02E+00

Table 10: Results of LEVO and other algorithms on composite test functions.

Function	LEVO		EVO		PSO		BA	
	Ave	std	ave	std	ave	std	Ave	std
F10	1.27E+01	2.22E-012	1.51E-04	3.82E-04	1.20E+02	1.32E+02	1.83E+02	1.17E+02
F11	2.32E-03	1.38E-03	1.46E+01	3.22E+01	1.63E+02	1.19E+02	4.87E+02	1.61E+02
F12	-9.93E-01	2.35E-02	1.75E+02	4.65E+01	3.63E+02	1.51E+02	5.88E+02	1.38E+02
F13	4.20E-01	2.39E-02	3.16E+02	1.30E+01	4.50E+02	1.58E+02	7.57E+02	1.60E+02
F14	3.06E+00	6.51E-02	4.40E+00	1.66E+00	1.75E+02	1.76E+02	5.42E+02	2.20E+02
F15	-8.35E-02	1.93E-01	5.00E+02	2.07E-01	9.02E+02	8.39E-01	8.19E+02	1.53E+02
Function	SMS		FPA		CS		GA	
	ave	std	ave	std	ave	std	ave	std
F10	7.77E+02	5.21E-12	3.37E-01	2.36E-01	1.10E+02	1.10E+02	1.15E+02	2.70E+01
F11	8.74E+02	9.72E+00	1.82E+01	3.08E+00	1.41E+02	9.28E+01	9.55E+01	7.16E+00
F12	9.61E+02	6.72E+01	2.24E+02	5.03E+01	2.90E+02	8.61E+01	3.25E+02	5.17E+01
F13	9.00E+02	1.99E-05	3.62E+02	5.40E+01	4.02E+02	9.82E+01	4.66E+02	2.96E+01
F14	7.41E+02	7.86E-01	1.02E+01	1.39E+00	2.13E+02	2.06E+02	9.04E+01	1.37E+01
F15	9.01E+02	8.44E-01	5.04E+02	1.16E+00	8.12E+02	1.92E+02	5.21E+02	2.80E+01

Performance of LEVO in large-scale problems: To verify the performance of the proposed LEVO algorithm in large-scale optimization problems, this subsection solves the 200-dimensional versions of the unimodal and multimodal test functions. LEVO algorithm is tested with two cases of parameters, as illustrated in Table 11. Results of the proposed LEVO algorithm are compared with PSO, SMS, BA, FPA, CS, FA, GA, and classic EVO (search agent=100, 5000 iteration). The results of unimodal test functions are reported in Table 12, and multimodal test functions are presented in Table 13.

According to the results in Tables 12,13, it is clear that LEVO (case1, 2) outperforms all the other algorithms in the majority of test cases. Also, LEVO (case 1) gives almost the same result as LEVO (case 2) although the search agent and iteration become lesser. These results show that LEVO avoids local optima, and resolves local optima stagnation in solving the challenging problem because of using levy flight.

Figures 4,5 illustrate the behavior of LEVO in solving small and large dimensions (Dim=30, Dim=200). It is obvious that the convergence behavior is almost the same in the case of increasing dimensions (convergence to optimal occurs in the same iteration, although the increase of dimensions).

Table 11: Parameters of two cases of LEVO.

	Case 1	Case 2
Search agent	30	100
Iteration	1000	5000

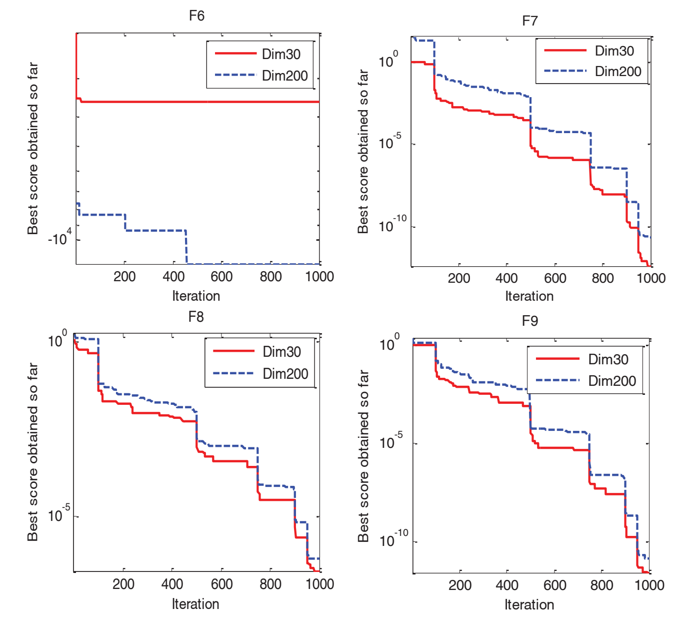


Figure 5: Behaviors of LEVO in small and large dimensions in multimodal test functions.

Table 12: Results of LEVO and other algorithms on unimodal test functions (200-dimensional).

Function	(case1) LEVO		(case2) LEVO		EVO		PSO		BA	
	Ave	std	Ave	std	Ave	std	Ave	std	Ave	std
F1	4.57E-11	6.34E-12	3.39E-11	3.87E-12	7.89E-07	1.10E-07	2.38E+01	1.17E+01	1.12E+03	2.07E+04
F2	5.57 E-06	2.92 E-07	5.09E-06	2.02E-07	5.31E+02	2.23E+02	2.38E+02	2.24E+01	3.84E+03	4.68E+02
F3	5.81 E-10	1.88 E-10	3.13E-10	6.56E-11	2.33E+03	5.07E+02	4.69E+03	5.04E+02	1.09E+03	4.75E+02
F4	2.21 E-06	2.62 E-07	1.64E-06	1.85E-07	3.06E+01	1.14E+00	4.01E+01	5.88E-01	6.57E+01	2.83E+00
F5	5.22 E-05	4.77 E-05	2.71E-06	2.49E-06	5.05E-02	1.44E-02	1.73E+01	4.01E+00	2.43E+00	1.28E-01
Function	SMS		FPA		CS		GA			
	Ave	std	Ave	std	Ave	std	Ave	Std		
F1	1.04E+03	4.24E-01	5.60E+01	3.27E+01	3.80E-05	1.85E-05	2.28E+02	1.87E+02		
F2	1.83E+03	1.22E-02	2.81E+02	6.94E+00	4.00E+02	8.66E-01	6.32E+03	1.09E+03		
F3	2.03E+03	3.78E-01	2.42E+01	8.54E+03	1.30E+01	6.34E+02	1.12E+01	3.99E+03		
F4	3.00E+02	2.30E-03	3.77E+01	2.46E+00	3.09E+01	1.69E+00	1.02E+02	2.53E+00		
F5	2.84E+01	1.99E-05	4.84E+00	1.54E+04	4.01E-01	8.71E-03	1.17E+02	6.02E+01		

Table 13: Results of LEVO and other algorithms on multimodal test functions (200-dimensional).

Function	(case1) LEVO		(case2) LEVO		EVO		PSO		BA	
	Ave	std	Ave	std	Ave	std	Ave	std	Ave	std
F6	-9.95E+03	817.14	-1.17E+04	7.89E+02	-4.44E+01	1.44E+03	-1.81E+01	4.96E+03	-2.56E+01	8.69E+02
F7	2.24 E-011	3.35 E-12	1.60E-11	2.10E-12	6.14E+02	6.68E+01	7.49E+02	2.43E+01	7.23E+02	1.01E+02
F8	6.17 E-07	3.52 E-08	5.26E-07	3.06E-08	2.31E+00	2.55E-01	1.52E+01	5.76E-01	1.82E+01	6.78E-02
F9	1.63 E-11	2.45 E-12	1.16E-11	1.15E-12	7.42E-03	6.51E-03	3.24E+03	1.37E+02	4.94E+03	2.68E+02
Function	SMS		FPA		CS		GA			
	Ave	std	Ave	std	Ave	std	Ave	std		
F6	-3.60E+01	8.77E-01	-4.58E+01	3.10E+03	-5.26E+01	1.56E+02	-2.87E+01	1.01E+03		
F7	4.80E+02	2.37E-01	7.03E+02	6.97E+01	5.42E+02	4.19E+01	1.65E+03	3.72E+01		
F8	1.73E+01	9.74E-02	1.75E+01	1.67E-01	1.77E+01	2.98E+00	2.04E+01	1.43E-01		
F9	4.80E+03	8.53E-01	1.81E+02	3.61E+01	1.19E-03	1.15E-03	3.31E+03	1.13E+02		

Conclusion

This work improved the behavior of a nature-inspired algorithm called EVO. This paper improves EVO by hybrid levy flight with the original EVO. Using levy flight has a better effect on the performance of the EVO algorithm, and this is because the levy walk makes a large jump of particles; this allows particles to Visit new sites, leads to local optima avoidance, and high exploration in search space. The proposed LEVO was benchmarked on fifteen test functions. These test functions are compared with algorithms described in the literature in terms of fitness improvement of the population, exploration, exploitation, and local optima avoidance. From the results, we can conclude that the proposed LEVO outperforms other algorithms on the majority of test functions. In addition, the performance of LEVO is tested in unimodal and multimodal large-scale problems. Results of EVO, GA, PSO, FA, SMS, BA, FPA, and CS (at a hundred search agents over 5,000 iterations) are compared with LEVO (at 30 search agents and 1,000 iterations). The results showed that LEVO outperforms all the other algorithms on the majority of test functions, although the search agent and iteration become less. Also, we can prove that the results of LEVO case 1 (30 search agents and 1000 iterations) give almost the same results as LEVO case 2 (at 100 search agents and 5000 iterations). Finally, we test the behavior of LEVO in small and large dimensions, and it turns out that the convergence behavior and convergence to optimal occur in the same iteration, although the dimensions increase. That makes us conclude that the proposed LEVO algorithm can be very effective for solving large-scale problems as well.

References

1. Azizi M, Aickelin U, A Khorshidi H, Baghalzadeh Shishehgarkhaneh M. Energy valley optimizer: a novel metaheuristic algorithm for global and engineering optimization. Sci Rep. 2023 Jan 5;13(1):226. doi: 10.1038/s41598-022-27344-y. PMID: 36604589; PMCID: PMC9816156.
2. Boussaïd I, Lepagnot J, and Siarry P, A survey on optimization metaheuristics. Inform Sci. 2013; 237: 82–117.
3. Holland JH, Reitman JS. Cognitive systems based on adaptive algorithms. ACM SIGART Bull. 1977; 49–49.
4. Colorni A, Dorigo M, Maniezzo V. Distributed optimization by ant colonies. In: Proceedings of the first European conference on artificial life. 1991; 134–42.
5. Eberhart RC, Kennedy J. A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science. 1995; 39–43.
6. Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Global Optim. 2007; 39:459–71.
7. Rajabioun R. Cuckoo optimization algorithm. Appl Soft Comput. 2011; 11:5508–18.
8. Yang X-S. Firefly algorithm, Levy flights and global optimization. In: Research and development in intelligent systems XXVI. Springer. 2010; 209–18.
9. Seyedali M, Mohammad MS, Andrew L. Grey wolf optimizer. Adv Eng Software. 2014; 69:46–61.

10. Faramarzi A, Heidarinejad M, Stephens B, Mirjalili S. Equilibrium optimizer: A novel optimization algorithm. Knowl.-Based Syst. 2020; 191: 105190.

11. MiarNaeimi F, Azizyan G, Rashki M. 'Horse herd optimization algorithm: A nature-inspired algorithm for high-dimensional optimization problems.' Knowl.-Based Syst. 2021; 213: 106711.

12. Trojovská E, Dehghani M, Leiva V. Drawer Algorithm: A New Metaheuristic Approach for Solving Optimization Problems in Engineering. Biomimetics. 2023; 8: 239.

13. Trojovský P, Dehghani M. A new bio-inspired metaheuristic algorithm for solving optimization problems based on walruses behavior. Scientific Reports. 2023; 13: 8775.

14. Elmanakhly DA. An Improved Equilibrium Optimizer Algorithm for Features Selection: Methods and Analysis. IEEE Access; 9, 2021.

15. Yilmaz S, Kucuksille E. A New Modification Approach on Bat Algorithm for Solving Optimization Problems. Applied Soft Computing Journal.

16. Chechkin A, Gonchar V, klafter J, Metzler R. Fundomentals of Lévy Flight processes. Adv Chem Phys. 2006; 133B: 439.

17. Fathi S, Makhlouf MAA, Osman E, Ahmed MA. An Energy-Efficient Compression Algorithm of ECG Signals in Remote Healthcare Monitoring Systems. IEEE Access. 2022; 10: 39129-39144.

18. ELmanakhly DA. BinHOA: Efficient Binary Horse Herd Optimization Method for Feature Selection: Analysis and Validations. IEEE ACCESS. 2022; 10.

19. Scharf I, Ovadia O. Factors influencing site abandonment and site selection in a sit-and-wait predator: a review of pit-building antlion larvae. J Insect Behav. 2006; 19:197–218.

20. Grzimek B, Schlager N, Olendorf D, McDade MC. Grzimek's animal life encyclopedia. Michigan: Gale Farmington Hills. 2004.

21. Yang XS. Nature-Inspired Metaheuristic Algorithms second edition.

22. Brown CT, Liebovitch LS, Glendon R, Lévy Flights in Dobe Ju/'hoansi Foraging Patterns. Hum Ecol. 2007; 35:129-138.

23. Cuevas E, Echavarría A, Ramírez-Ortegón MA. An optimization algorithm inspired by the States of Matter that improves the balance between exploration and exploitation. Applied Intelligence. 2014; 40(2): 256-272.

24. Yang XS. A new metaheuristic bat-inspired algorithm. In: Nature inspired cooperative strategies for optimization (NICSO 2010). Springer; 2010; 65–74.

25. Yang XS. Flower pollination algorithm for global optimization. In: Unconventional computation and natural computation. 2012; 240–9.

26. Yang XS, Deb S. Cuckoo search via Lévy flights. In: World congress on nature & biologically inspired computing. 2009. NaBIC 2009; 210–4.

How to cite this article: Shikoun NH, Fathi IS. Improved Energy Valley Optimizer with Levy Flight for Optimization Problems. IgMin Res. April 18, 2024; 2(4): 245-254. IgMin ID: igmin172; DOI: 10.61927/igmin172; Available at: igmin.link/p172