International Journal of Engineering & Technology, 7 (4) (2018) 4570-4576

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 22726
Research paper

An improved African buffalo optimization algorithm using chaotic map and chaotic-levy flight

Chinwe Peace Igiri [1] *, Yudhveer Singh [1], Deepshikha Bhargava [2]

[1] Amity Institute of Information Technology, Amity University Rajasthan, India
[2] Prof. School of Computer Science, University of Petroleum and Energy Studies, Dehradun, India
**Corresponding author E-mail: chynkemdirim@gmail.com*

Abstract

Optimization is ever-growing research that cuts across all walks of life. Many popular metaheuristic algorithms have metamorphosed into numerous variants in search of the sophisticated kernel for optimal solution. The African Buffalo optimization (ABO) algorithm is one of the fastest metaheuristic algorithms. This algorithm is inspired by the alarm and alert calls of African buffaloes during their foraging and defending activities. The present study investigates the strengths and weaknesses of ABO and proposes two improvement strategies: Chaotic ABO (CABO) and chaotic-levy flight ABO (CLABO). The results are validated with ten benchmark optimization problems and compared with other metaheuristic algorithms in the literature. Further, the CABO and CLABO algorithms are ranked first and second, respectively. This proves the superiority of the proposed improved algorithms over others under this study. Finally, the improved chaotic ABO would be utilized for optimizing industrial scheduling for oil and gas in our future work.

Keywords: *Chaotic Optimization; African Buffalo Optimization; Levy-Flight; Non-Linear Optimization; Meta-Heuristics.*

1. Highlights

- Chaos and levy flight have been introduced to ABO.
- The effectiveness of the two new variants (CABO and CLABO), have been tested with ten benchmark functions.
- The result shows that both CABO and CLABO have relatively high convergence speed in comparison with other metaheuristic algorithms.
- Chaos is a better improvement strategy for ABO algorithm than combination of chaos and levy flight.

2. Introduction

The increasing relevance of nature-inspired computation as a soft computing technique is the driving force for the seemingly endless search for optimal solution in various problem domains. Optimization has been a subject of concern for research. It is a converging point for computer science, mathematics, operations research, economics, etc. In operational research, combinatorial optimization problem refers to the determination of optimal solution to perform a collection of tasks by a number of agents at minimal cost, time and resources [1]. Conventional optimization algorithms such as Augmented Lagrangian methods [2], Jacobian optimization method [3, 4] etc., incur high computational cost and are inefficient for most optimization problems. Literature reveals that nature-inspired algorithms are better alternatives. Algorithms such as the ant colony optimization (ACO) [5], ABO [6] migrating bird optimization [7], etc. have been proven more effective than conventional algorithms. Although these algorithms are more efficient than their conventional counterparts, there yet exist certain drawbacks. Rather than developing new ones, it is better to improve the efficiency of the existing ones [8].

The ABO algorithm, developed by Odili and Kahar [9], belongs to the class of metaheuristic population-based nature-inspired algorithms. ABO like Bat algorithm [10] employs the sense of sound, which varies from the breeding parasitism of cuckoo search [11], flashing and attraction of firefly [12], to flocking characteristics of particle swarm optimization (PSO). Odili et al. [9] claimed that ABO was developed to address the problem of premature convergence as well balance exploration and exploitation by utilizing few parameters. However, this claim is not justifiable in all optimization problems. ABO details would be discussed subsequently in this study. In terms of application domain, ABO has been utilized to solve traveling salesman problem (TSP) [9]. According to that study, ABO outperformed genetic algorithm (GA), ant colony optimization (ACO), honey bee mating optimization (HBMO), simulated annealing (SA), and many more [9]. The algorithm has also been tested on numerical functions with better result than GA and improved GA [13]. However, it was also found that chaotic gray-coded GA yielded better result than ABO [13]. In fact, that was the motivation to improve ABO using chaotic optimization. Further, ABO has shown better performance than randomization insertion algorithm (RIA) in asymmetric TSP [14]. Additionally, it has outperformed PSO, GA, ACO, and bacterial foraging optimization (BFO) in turning PID controller parameters [15]. Recently, ABO has been utilized to solve budget constraint maximal covering location (BCMCL) using the binary African buffalo optimization (BABO) algorithm [16].

Although most metaheuristic algorithms in basic state could solve several optimization problems, they have some limitations. One of them is premature convergence, which could result in a non-feasible solution [8]. To a large extent, hybridization and/or use of sophisticated randomization technique could improve such algorithm performance in most optimization problems. Many studies show that modified algorithms outperform their basic counterparts; however, there are some exceptional cases. Rao et al. [17]

Copyright © 2018 Chinwe Peace Igiri et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

developed an improved hybrid (ACO and GA) algorithm for industrial production operation. Also, Wang et al. [18], designed a hybrid GA and DE for joint replenishment and location inventory problem in a three-level supply chain. Li et al. [19] formulated hybrid genetic-simulated annealing algorithm (HGSAA) for e-supply chain environment. HGSAA outperformed GA in terms of computing time, optimal solution, and computing stability. Yu et al. [20] also formulated an adaptive hybrid of PSO and DE for global optimization in scientific and engineering fields. Furthermore, Duan et al. [21] proposed a hybrid optimization algorithm of finite state method and GA to solve the crude oil scheduling problem. These studies among others justify the need for algorithm improvement. The remaining part of this study is organized as follows: section 2 is the description of basic ABO, and the improvement strategies are discussed in section 3. Section 4 is the result and discussion while the study is summarized with conclusion and recommendations in section 5.

3. Basic ABO

ABO, belonging to the class of swarm intelligence, was developed by Odili et al. [9]. This metaheuristic algorithm models the foraging and defending behavior of African buffaloes. The unique features of these animals include extensive memory capacity, communal lifestyle, and democratic lifestyle [9]. They utilize 'waaa' and 'maaa' sounds to communicate danger and safety, respectively. Thus, their organizational lifestyle could be mapped to these unique characteristics [22, 9].The "waaa" sound is denoted by w_i, the "maaa" denoted by m_i, while the learning parameters are denoted by l_1 and l_2. Other parameters are global maximum ($b_o$gmax), the personal maximum (b_o pmax.b) positions. The basic ABO is controlled by two equations, namely democratic Eq.1 and location update Eq.2 equations. Algorithm 1 shows the basic ABO. The algorithm sub- tracts the "waaa" value (w_i) asking the animals to explore the search space from the maximum vector ($b_o$gmax and $b_o$pmax.a) which is further multiplied by the learning parameters (l_1 and l_2) [9]. The result is supplied by the "maaa" (m_i) value, this indicates that they herds should remain in that location and continue grazing.

$$m_{i+1} = m_i + l_1(b_o gmax - w_i) + l_2(b_o pmax.i - w_i) \quad (1)$$

$$w_{1+1} = \left(\frac{w_i + m_i}{\lambda}\right) \quad (2)$$

Algorithm 1 ABO [9] Step1. Randomly initialize buffaloes within the search area
Step2. Update buffaloes' exploitation with equation Eq. 1
Step3. Update the location of buffaloes with Eq. 2
Step4. If equation Eq. 1 and Eq. 2 is updating, proceed to Step 5. Otherwise, return to step 1
Step5. If stopping criteria is reached, proceed to Step 6, else return to Step 2
Step6. Output best solution

3.1. Strengths and weaknesses of basic ABO

From the foregoing, one could say that ABO has shown good solutions with better speed in TSP, benchmark numerical functions, etc. [9], [14], [13]. Also, the strength of ABO has been proven by low relative percentage error obtained in multi-modal and unimodal functions [13]. The speed of this algorithm is due to the few parameters that control the algorithm [15].
However, since there is no best algorithm for all optimization problems [8], ABO like other basic metaheuristic algorithms exhibit a number of weaknesses, such as inefficient for multivariable optimization problems [13] and non-linear constrained optimization problems. Besides, it would not be out of place to say that the kernel (mathematical model) of ABO is so simplified that it does not account for the comprehensive characteristics of African Buffaloes in their foraging and defending activities. To illustrate, the buffaloes settle on the green pasture at the sound of 'maaa' call [9] as represented in the kernel. This could be justified by the lack of crossover in the updating equations. The herds are intelligent enough to know that there could be greener pasture based on their previous grazing experiences. However, the mathematical formulation of this algorithm does not account for this feature. This could result in converging at near optimum solution in most problems. Also, it utilizes simple randomization techniques, unlike CS and FA that employ levy flight and Gaussian technique, respectively [8]. Strictly speaking, these weaknesses are the motivation for this study. Thus, the proposed study is aimed at enhancing the kernel of the algorithm to account for their extensive memory capacity feature. The summary of the related works in ABO is illustrated in Table 1.

4. ABO Improvement

ABO is basically controlled by Eqs.1 and 2 which are referred to as democratic and location update equations, respectively, as mentioned earlier. However, these two equations did not account for the exceptional memory capacity of African buffaloes. In other words, the herd identifies a grazing land and relaxes without exploring possible greener pastures based on previous experience. Thus, possible reason for pre-mature convergence observed in most optimization problems. Also, the λ in location update equation Eq.2 is a simple random number generator, this implies that the search is aimless within the search space resulting in relatively inefficient solution and/or premature convergence as the case may be. However, employing chaos and levy distribution properties could be a possible solution to these problems.

4.1. Chaotic map

A system is said to be chaotic if it exhibits a kind of random deterministic behavior in a bounded but non-converging search space [23]. Many stochastic optimization problems are trapped in local optima. However, literature show that chaotic sequence or map could be used to deal with local optima problem [24]. Some examples of such algorithms are GA, DE, PSO, FA, ACO, SA, imperialistic competitive algorithm, charged system search optimization, and big- bang big crunch optimization, etc. as referenced therein [25], [26]. Basically, chaotic optimization is the use of chaotic sequence instead of random variables in an optimization algorithm.
There are variations of one-dimensional chaotic maps including, logistic, iterative, sinusoidal, sin, circle, chebyshev, intermittency, singer, sawtooth, pieceswise, tent, and liebovtch [27]. However, this study considers logistic map for ABO randomization. Eq. 3 represents the logistic function [28], [26].

$$\mu_{m+1} = \beta.\mu_m(1 - \mu_m) \quad (3)$$

4.2. Levy flight

Levy walk is a description of diffusion pattern of organisms such that searching is concentrated at the location of potential solution

Table 1: ABO Related Works

	Description	Method/Ref	Result	Drawback
1	Solve TSP	ABO [9]	Outperformed GA, HBM, SA, etc.	Evaluation restricted to TSP
2	Evaluation of numerical functions	ABO [13]	Outperformed GA and improved GA	Outperformed by chaotic Grey coded GA
3	Evaluation of Asymmetric TSP	ABO [14]	Outperformed RIA	Insufficient comparative analysis
4	Turning PID controller parameters	ABO [15]	Outperformed PSO, ACO, and BFO	Evaluation restricted to PID controller
5	Solve (BCMCL) problem	BABO [16]	Competitive result with other algorithms	Only approximately 67% success rate

[29]. Levy flight foraging hypothesis estimates the migration from less-resource to more-resource environment which in turn results in optimal search [30]. Animals with high memory capability employs this model to explore their search space [30]. The theory of optimal foraging is an extension of Levy flight foraging hypothesis, which states that organism pay closer attention to the optimal solution location rather than aimless search within the search space. Levy flights are random walks whose step length is drawn from the Levy distribution, often in terms of a simple power-law formula

$L(\zeta) \sim \zeta^{-1-\alpha}$ where $0 < \alpha < 2$ is an index. Levy flight could ben mathematically represented as

$$L(\zeta,\omega,\psi) = \begin{cases} \sqrt{\frac{\omega}{2\pi}}\ exp\left[-\frac{\omega}{2(\zeta-\psi)}\right]\frac{1}{(\zeta-\psi)^{3/2}}, & 0 < \psi < \zeta < \infty \\ 0 & otherwise \end{cases} \quad (4)$$

Where $\psi > 0$ is a minimum step and $\omega$is a scale parameter. Ideally as $\zeta \to \infty$, then

$$L(\zeta,\omega,\psi) \approx \sqrt{\frac{\omega}{2\pi}}\frac{1}{\zeta^{3/2}}. \quad (5)$$

Mantegna algorithm [8] would be utilized for levy flight implementation in this study. Thus, the step length ζ would be calculated by

$$\zeta = \frac{\ell}{|\kappa|^{1/\alpha}}, \quad (6)$$

Where ℓ and κ are drawn from normal distributions (Yang, 2014). In other words,

$$\ell \sim N\left(0,\rho_\ell^2\right), \kappa \sim N\left(0,\rho_\ell^2\right),$$

Where

$$\rho\ell = \left\{\frac{\Gamma(1+\alpha)\ sin(\pi\alpha/2)}{\Gamma[1+\alpha/2]\ \alpha 2^{(\alpha-1)/2}}\right\}^{1/\alpha}, \rho\kappa = 1\,. \quad (7)$$

Readers are referred to [8] for details on Levy flight.

4.3. Proposed improved ABO algorithm

The improvement of ABO considers in two ways namely: Chaotic ABO (CABO) and Chaotic-Levy Flight ABO (CLABO).

4.3.1. CABO

In this case, the learning terms of the democratic equation Eq.(1) is multiplied by a suitable chaotic sequence. This is to enhance exploration of the algorithm and guide against premature convergence. That is, given a chaotic map

$$\mu_{i+1} = f(\mu_i) \quad (8)$$

The democratic and exploitation equation have been modified as shown in Eqs. 9 and 10

$$m_{i+1} = m_i + l_1\mu.*((b_o gmax - w_i) + l_2\mu.*(b_o pmax.i - w_i) \quad (9)$$

$$w_{1+1} = \left(\frac{w_i + m_i}{\lambda}\right) + \mu \quad (10)$$

Where.* is element-wise multiplication of vectors. The step by step implementation is shown in Algorithm 2

The main property of chaos is that a slight variation in underlying population drastically yield unique outcome at each iteration. This characteristic enables the system to generate dynamic values at every iteration (search), which consequently enhance the quality of the solution within a shortest possible time. In this proposed CABO, an initial value of 0.01 and parameter value of 4 for the logistic map is used, as shown in Eqs.9 and 10. This enables the buffaloes (solutions) generate an efficient search mechanism that could escape local optima entrapment. The steps are illustrated in Algorithm 2 .First, the Buffaloes are randomly initialized within the search space. Then, the chaotic sequence is generated. Next, the exploitation and democratic process is executed using chaotic map and chaotically enhanced parameters (l_1, l_2) respectively. These processes are carried out iteratively until the stopping criteria is reached. In other words at each iteration, the fitness value of the new solution is evaluated to select the best solution. The maximum number of iteration is set to 100. Ten non-linear benchmark problems are utilized for the performance validation of the proposed CABO. The flow diagram of the proposed CABO is shown in fig. 1, the red colour highlight captures modification of the basic ABO to CABO.

4.3.2. CLABO

Here, the aim is to locate the "greenest pasture" based on levy foraging hypothesis [30]. This would result in modification of both the democratic and location update equations as shown in Eqs. 11 and 12.

$$m_{i+1} = m_i + l_1\mu.*((b_o gmax - w_i) + l_2\mu.*(b_o pmax.i - w_i) \quad (11)$$

$$w_{i+1} = \left(\frac{w_i + m_i}{\lambda}\right) + \mu.*L(\zeta) \quad (12)$$

where.* is element-wise multiplication of vectors, μ is the chaotic map and L (ζ) is the Levy flight. Algorithm 3 shows the step-wise implementation of the improved CLABO.

During this process, the chaotic sequence is also generated as in the case of CABO. However, since the buffaloes possess the capability to recall the possibility of a "greener pasture" due to their exceptional memory capacity. Therefore, Levy flight distribution process is incorporated into the search process. This search mechanism enables the buffaloes intelligently recall that there could be "greener pasture" based on the levy foraging hypothesis proposed by Gautestad et al. [30]. Thereby inducing concentrated search in possible location of better solution within the shortest possible time also. The optimization steps of the proposed CLABO is presented in Algorithm 3. First, the buffaloes are randomly initialized within the search region. Then, the chaotic sequence is generated, followed by the Levy distribution process using Mantegna's algorithm. Next, the exploitation and democratic processes are executed via chaotic and Levy flight process as seen in Eqs.11 and 12. The fitness of the buffaloes are evaluated using various non-linear benchmark functions with maximum of 100 iterations.

Algorithm 2 CABO

Step1. Randomly initialize buffaloes within the search area Step2. Generate chaotic sequence or map
Step3. Update buffaloes' exploitation with Eq. 10 enhanced by chaotic map
Step4. Update the location of buffaloes with Eq. 9
Step5. If Eqs. 10 and 9 is updating, proceed to Step 6. Otherwise, return to step 1
Step6. If stopping criteria is reached, proceed to Step 7, else return to Step 3
Step7. Output best solution

Algorithm 3 CLABO

Step1. Randomly initialize buffaloes within the search area Step2. Generate chaotic sequence

Step3. Update buffaloes' exploitation with equation Eq. 11 enhanced by chaotic map
Step4. Perform Levy flight via Mantegna's algorithm
Step5. Update the location of buffaloes with Eq. 12 enhanced by chaotic and Levy flight
Step6. If equation Eqs.11 and 12 are updating, proceed to Step 7. Otherwise, return to step 1
Step7. If stopping criteria is reached, proceed to Step 8, else return to Step 3
Step8. Output best solution

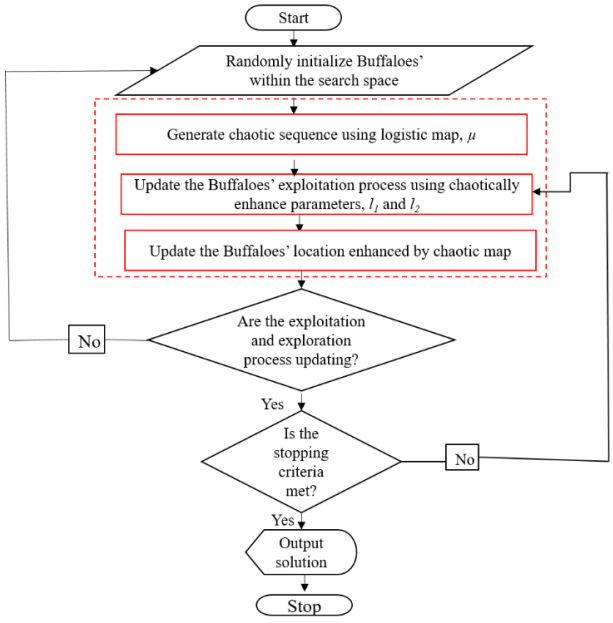


Fig. 1: CABO Illustration Flow Chart.

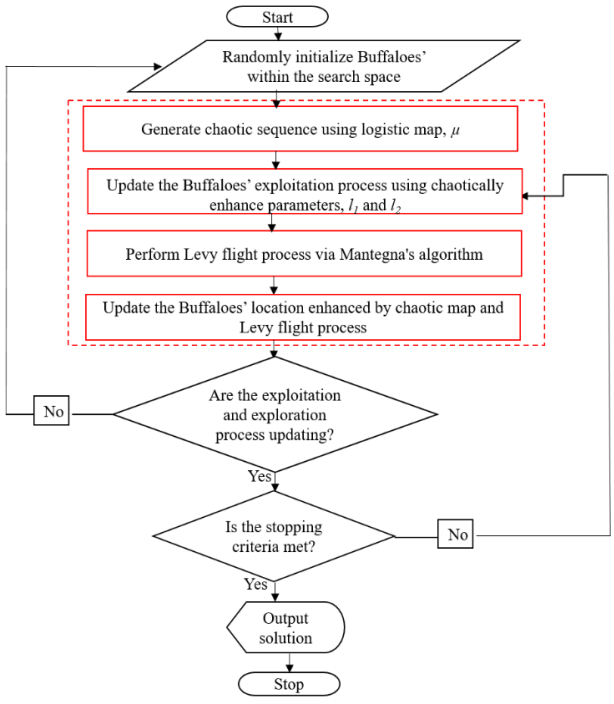


Fig. 2: CLABO Illustration Flow Chart.

5. Results and discussion

5.1. Numerical experiments

Ten benchmark functions are used to examine the performance of the improved algorithms. These include

1) Sphere (F1),

$$\varphi(x) = \sum_{i=0}^{n} x_i^2, \quad x \in [-10,10] \tag{13}$$

2) Matyas (F2),

$$\varphi(x) = 0.26\,(x_1^2 + x_2^2) - 0.48 x_1 x_2, \; x \in [-10,10], 0 \tag{14}$$

3) Six Hump Camel Back (F3),

$$\varphi(x) = 4x_1^2 - 2.1x_1^4 + 1/3x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4, x \in [-5,5], -1.03163 \tag{15}$$

4) Easom (F4),

$$\varphi(x) = -cos(x_1)\,cos(x_2)\,exp(-(x_1-\pi)^2 - (x_2-\pi)^2), x \in [-100,100], -1 \tag{16}$$

5) Ackley (F5),

$$\varphi(x) = -20\,exp\left[-0.2\sqrt{0.5(x_1-x_2)}\right] - exp[0.5(cos\,2\pi x_1 + cos\,2\pi x_2)] + exp(1) + 20, \; x \in [-600,600], 0 \tag{17}$$

6) Zakharov (F6),

$$\varphi(x) = \sum_{i=1}^{n} x_i^2 + (\sum_{i=1}^{n} 0.5 i x_i)^2 + (\sum_{i=1}^{n} 0.5 i x_i)^4, \; x \in [-5,10], 0 \tag{18}$$

7) Schaffer (F7),

$$\varphi(x) = 0.5 + \frac{sin(x_1 - x_2)^2 - 0.5}{[1+0.001\,(x_1^2+x_2^2)]^2}, \; x \in [-100,100], 0 \tag{19}$$

8) Bochachvesky 1(F8),

$$\varphi(x) = x_1^2 + 2\,x_2^2 - 0.3\,cos(3\pi x_1) - 0.4\,cos(4\pi x_{12}) + 0.7, \; x \in [-100,100], 0 \tag{20}$$

9) Griewank (F9),

$$(x) = 1 + \frac{1}{4000}\sum_{i=1}^{n} x_1^2 - \prod_{i=1}^{n} cos\left(\frac{x_i}{\sqrt{i}}\right), x \in [-600,600], 0 \tag{21}$$

10) Rastrigin (F10) functions

$$\varphi(x) = 10n + \sum_{i=1}^{n}[x_i^2 - 10\,cos(2\pi x_i)], \quad x \in [-5.2,5.2], 0 \tag{22}$$

The choice of these benchmark functions capture polynomials, transcendental, multidimensional and multi-modal functions. The numerical experimentation seeks to reveal the convergence of the improved algorithms and comparative analysis with respect to the existing algorithms. The existing algorithms used for comparison include; These algorithms include PSO [11], improved cuckoo search (ICS) [31], advanced cuckoo search (ACS) [31], gravitational search algorithm (GSA) [32], enhanced opposition-based firefly algorithm (EOFA) [32], FA [32], [12], and parallel migrating genetic algorithm (PMGA) [11].

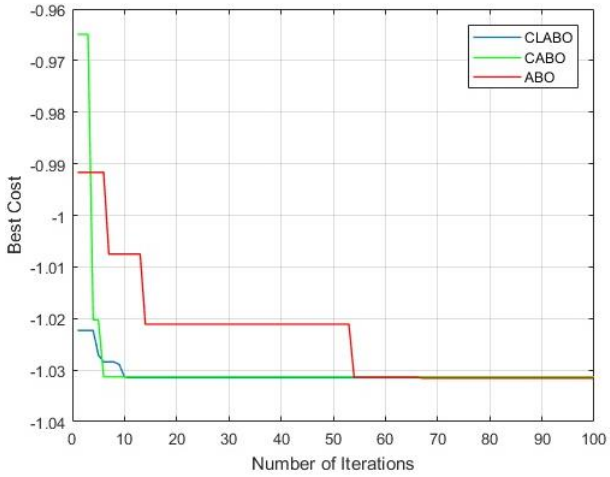


Fig. 3: Illustration of Convergence of Six-Hump Camel Back Function.

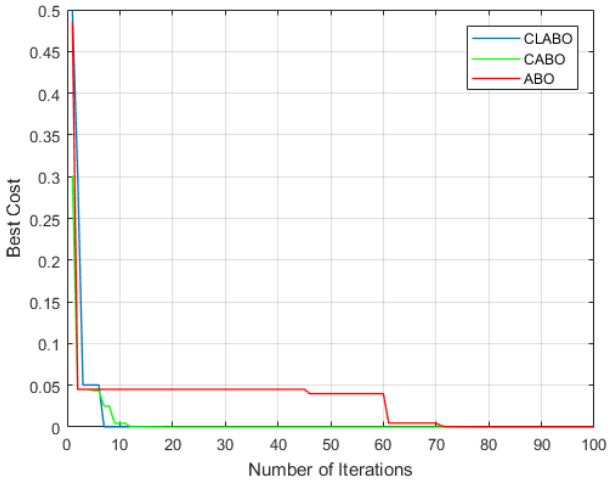


Fig. 4: Illustration of Convergence of Schaffer Function.

5.2. Performance evaluation

The proposed CABO and CLABO have been evaluated using ten benchmark functions. The performances of CABO and CLABO have been compared with that of the basic ABO (which serves as control) and other algorithms in the literature as earlier stated.

Ideally, the efficiency and robustness of an algorithm is a function of its capability to converge towards global minimum with fewer iterations. Thus, the average number of evaluation (ANOFE) and the value of the objective function are selected as criteria for performance metric in this study.

Further, 30 independent runs were carried out on the ten benchmark functions with 100 maximum number of iterations. The statistical values of mean, minimum (best), maximum (worst), and standard deviation have also been taken, as illustrated in Tables 2 and 3. The ranking was used to summarize the comparative performance evaluation, as shown in Table 4. Also, the mean performance was used as a comparative standard since it is the common statistical value available for all other algorithms in literature. The two parameters, l_1 and l_2 that control ABO algorithm are both set to 0.8 across both basic and proposed improved variants.

As mentioned earlier, the performance of the algorithms on the benchmark functions are represented in Tables 2 and 3. The best algorithm is highlighted with bold for emphasis and clarity. Also, CABO, CLABO, and ABO ranked first, second and third, respectively in comparison to other algorithms as seen in Table 4. More so, CABO and CLABO converged at global minimum for functions F7, F8, F9, and F10 with relatively fewer iterations than other algorithms as shown in Table 3. Surprisingly, ABO outperformed all other algorithms on Six Hump Camel Back and Easom function whose global minimum is -1. This suggests that chaos and levy flight have negative impact on these test functions. Additionally, only two convergence graphs, Fig. 3 and Fig. 4 have been selected for visual perception and at the same time ensure a simplified report. These convergence plots illustrate the performance of the proposed improved variants of ABO over the basic ABO, thus revealing the novelty of the proposed improvement strategies.

6. Conclusion and recommendations

Basic ABO, like other metaheuristic algorithms, is effective but with some limitations. Its efficiency could be ascribed to a few parameters which enable it to obtain a good solution with relatively few iterations. However, the limitations are the tendency of local minimum entrapment and inefficient search exploration. A two-level improvement using chaos and levy flight has been done. Specifically, the exploration (democratic) and exploitation equations have been independently enhanced: First, with only chaos, and second, with a combination of chaos and levy flight.

In the light of the foregoing, the following conclusion could be drawn from the proposed improved ABO variants. Firstly, CABO and CLABO are efficient since optimal or near optimum solutions were realized with relatively few ANOFA. Secondly, CABO and CLABO are robust not only because of the 100% success rate but also the high-quality objective values obtained across all the benchmark functions under study. Thirdly, chaos has more impact on the search efficiency of basic ABO than the combination of chaos and levy flight. Finally, the optimization problems should first be implemented using basic algorithms before applying the improved variants.

Ultimately, chaos has been proven to be the best enhancement strategy to alleviate computational effort and improve solution quality of basic ABO. This study could be extended to constrained as well as real-world optimization problems such as process scheduling, engineering design, among others.

Table 2: Algorithms Performance Evaluation Using Benchmark Functions

Functions	Algorithms	Iteration No.	Best	Worst	Mean	STD
	CLABO	100	6.7494E-56	2.6E-54	5.56E-55	5.58E-55
	CABO	100	1.60E-57	1.03E-55	2.37E-56	2.50E-56
F1	ABO	100	2.84E-13	3.28E-11	3.28E-12	8.49E-12
	ICS[31]	1000	N/A	N/A	1.85	1.85
	ACS[31]	1000	N/A	N/A	4.71	5.87
	CLABO	100	1.30E-60	6.50E-56	1.07E-56	1.48E-56
	CABO	100	2.51E-63	3.00E-56	2.12E-57	5.751E-56
F2	ABO	100	2.29E-14	3.73E-12	5.17E-13	8.08E-13
	GSA [32]	1000	4.73E-09	9.11E-07	1.35E-07	N/A
	EOFA [32]	1000	1.59E-40	8.06E-36	1.45E-36	N/A
	FA [32]	1000	1.30E-05	0.58	0.04	N/A
	CLABO	100	-1.0316	-1.0249	-1.0300	0.0017
	CABO	100	-1.0316	-1.0301	-1.0311	0.0004
F3	ABO	100	-1.0316	-1.0311	-1.0315	0.0002
	ICS[31]	1000	N/A	N/A	3.3630	4.1773
	ACS[31]	1000	N/A	N/A	1.3234	2.1494
F4	CLABO	100	-0.9997	-0.9677	-0.9888	0.0095

	CABO	100	-0.9999	-0.9839	0.9956	0.0043
	ABO	100	-0.9999	-0.9909	-0.9974	0.0024
	ICS[31]	1000	N/A	N/A	1.7999	1.8598
	ACS[31]	1000	N/A	N/A	1.7495	1.7534
	CLABO	100	8.88E-16	8.88E-16	8.88E-16	4.01E-31
	CABO	100	8.88E-16	8.88E-16	8.88E-16	4.01E-31
F5	ABO	100	1.32E-06	7.53E-06	1.80E-06	1.44E-06
	ICS[31]	1000	N/A	N/A	3.6053	4.5822
	ACS[31]	1000	N/A	N/A	0.0288	2.3097
	CLABO	100	3.59E-57	1.36E-13	5.24E-36	2.53E-14
	CABO	100	4.28E-55	1.51E-40	5.04E-42	2.53E-41
F6	ABO	100	1.68-E11	2.75E-08	2.56E-08	2.56E-09
	GSA [32]	1000	6.80E-35	73.96	56.79	N/A
	EOFA [32]	1000	5.84E-35	1.60E-29	1.92E-30	N/A
	FA [12]	1000	708.22	3.92E+09	5.47E+08	N/A

Table 3: Algorithms Performance Evaluation Using Benchmark Functions Contd.

Functions	Algorithms	Iteration No.			Objective Value		
		Best	Worst	Mean	Best	Worst	Mean
	CLABO	42	50	46	0	0	0
	CABO	41	67	44	0	0	0
F7	ABO	100			4.91E-11	4.37E-02	2.91E-03
	PMGA [11]	10,000			N/A		12.63
	PSO [20]	10,000			N/A		5.30E-08
	FA [12]	10,000			N/A		6.63E-15
	CLABO	42	46	44	0	0	0
	CABO	41	67	44	0	0	0
F8	ABO	100			1.06E-11	6.22E-08	7.38E-09
	GSA [32]	1000			1.09E-06	1.05E-04	2.06E-05
	EOFA [32]	1000			0	2.22E-16	4.88E-17
	FA [32]	1000			2.10E-04	3.35	0.55
	CLABO	38	56	43	0	0	0
	CABO	35	72	48	0	0	0
F9	ABO	100			1.48E-12	0.05	0.01
	GSA [32]	1000			6.8E-06	0.03	1.78E-03
	EOFA [32]	1000			0	5.55E-17	1.78E-17
	FA [32]	1000			446.74	686.12	592.46
	CLABO	42	60	50	0	0	0
	CABO	42	65	49	0	0	0
F10	ABO	100			1.85E-10	1.0839	4.52E-02
	GSA [32]	1000			15.99	53.79	34.5
	EOFA [32]	1000			0	3.19	0.99
	FA [32]	1000			353.45	429.4	394.46

Table 4: Algorithms Performance Evaluation Ranking

Algorithm/Function	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Sum/n*10	Ranking
CABO	10	10	9	9	10	10	10	10	10	10	0.98	1
CLABO	9	9	8	8	10	9	9	10	9	9	0.9	2
ABO	8	7	10	10	8	7	8	8	6	8	0.8	3
ICS[31]	7	-	6	6	6	-	-	-	-	-	0.625	7
ACS[31]	6	-	7	7	7	-	-	-	-	-	0.675	6
EOFA [32]	-	8	-	-	-	8		7	8	7	0.76	4
GSA [32]	-	6	-	-	-	6		6	7	6	0.62	8
FA [32]	-	5	-	-	-	5		5	5	-	0.5	10
PMGA [11]	-	-	-	-	-	-	5	-	-	-	0.5	10
PSO [11]	-	-	-	-	-	-	6	-	-	-	0.6	9
FA [12]	-	-	-	-	-	-	7	-	-	-	0.7	5

n = number of benchmark functions.

References

[1] A Bagheri, Mostafa Zandieh, Iraj Mahdavi, and Mehdi Yazdani. An artificial immune algorithm for the flexible job-shop scheduling problem. *Future Generation Computer Systems*, 26(4):533–541, 2010. https://doi.org/10.1016/j.future.2009.10.004.

[2] EG Birgin, G Haeser, and a Ramos. Augmented lagrangians with constrained sub problems and convergence to second-order stationary points. *Optimization Online*,2016.

[3] Vahid Beiranvand, Warren Hare, and Yves Lucet. Best practices for comparing optimization algorithms. *Optimization and Engineering*, 18(4):815–848, 2017. https://doi.org/10.1007/s11081-017-9366-1.

[4] Mohui Jin, Xianmin Zhang, Zhou Yang, and Benliang Zhu. Jacobian- based topology optimization method using an improved stiffness evaluation. *Journal of Mechanical Design*, 140(1):011402, 2018. https://doi.org/10.1115/1.4038332.

[5] Francesco Orciuoli, Mimmo Parente, and Autilia Vitiello. Solving the shopping plan problem through bio-inspired approaches. *Soft Computing*, 20(5):2077–2089, 2016. https://doi.org/10.1007/s00500-015-1625-5.

[6] Julius Beneoluchi Odili, Mohammad Nizam Kahar, and a Noraziah. Solving traveling salesman's problem using African buffalo optimization, honey bee mating optimization & linkerninghan algorithms. *World Applied Sciences Journal*, 34(7):911–916, 2016.

[7] Biao Zhang, Quan-ke Pan, Liang Gao, Xin-li Zhang, Hong-yan Sang, and Jun-qing Li. An effective modified migrating bird's optimization for hybrid flowshop scheduling problem with lot streaming. *Applied Soft Computing*, 52:14–27, 2017. https://doi.org/10.1016/j.asoc.2016.12.021.

[8] Xin-She Yang. *Nature-inspired optimization algorithms*. Elsevier, 2014.

[9] Julius Beneoluchi Odili, Mohd Nizam Mohmad Kahar, and Shahid Anwar. African buffalo optimization: a swarm-intelligence technique. *Procedia Computer Science*, 76:443–448, 2015. https://doi.org/10.1016/j.procs.2015.12.291.

[10] Xin-She Yang and Amir Hossein Gandomi. Bat algorithm: a novel approach for global engineering optimization. *Engineering Compu-*

tations, 29(5):464–483, 2012. https://doi.org/10.1108/02644401211235834.
[11] Xin-She Yang and Suash Deb. Cuckoo search via lévy flights. In *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, pages 210–214. IEEE, 2009.
[12] Xin-She Yang. Firefly algorithms for multimodal optimization. In *International symposium on stochastic algorithms*, pages 169–178. Springer, 2009. https://doi.org/10.1007/978-3-642-04944-6_14.
[13] Julius Beneoluchi Odili and Mohd Nizam Mohmad Kahar. Numerical function optimization solutions using the African buffalo optimization algorithm (abo). *British Journal of Mathematics & Computer Science*, 10(1):1–12, 2015.
[14] Julius Beneoluchi Odili, Mohd Nizam Mohmad Kahar, Shahid Anwar, and Mohammed Adam Kunna Azrag. A comparative study of African buffalo optimization and randomized insertion algorithm for asymmetric travelling sales clerk's problem. In *Software Engineering and Computer Systems (ICSECS), 2015 4th International Conference on*, pages 90–95. IEEE, 2015.
[15] Julius Beneoluchi Odili, Mohd Nizam Mohmad Kahar, and A Noraziah. African buffalo optimization algorithm for tuning parameters of a PID controller in automatic voltage regulators.
[16] Boris Almonacid, Juan Reyes-Hagemann, Juan Campos-Nazer, and Jorge Ramos-Aguilar. Selecting a biodiversity conservation area with a limited budget using the binary African buffalo optimization algorithm. *IET Software*, 2017.
[17] Yunqing Rao, Dezhong Qi, and Jinling Li. An improved hierarchical genetic algorithm for sheet cutting scheduling with process constraints. *The Scientific World Journal*, 2013, 2013.
[18] Lin Wang, Hui Qu, Tao Chen, and Fang-Ping Yan. An effective hybrid self-adapting differential evolution algorithm for the joint replenishment and location-inventory problem in a three-level supply chain. *The Scientific World Journal*, 2013, 2013.
[19] Yanhui Li, Hao Guo, Lin Wang, and Jing Fu. A hybrid genetic-simulated annealing algorithm for the location-inventory-routing problem considering returns under e-supply chain environment. *The Scientific World Journal*, 2013, 2013.
[20] Xiaobing Yu, Jie Cao, Haiyan Shan, Li Zhu, and Jun Guo. An adaptive hybrid algorithm based on particle swarm optimization and differential evolution for global optimization. *The Scientific World Journal*, 2014, 2014.
[21] Qian-Qian Duan, Gen-Ke Yang, and Chang-Chun Pan. A novel algorithm combining finite state method and genetic algorithm for solving crude oil scheduling problem. *The Scientific World Journal*, 2014, 2014.
[22] David Sloan Wilson. Altruism and organism: Disentangling the themes of multilevel selection theory. *The American Naturalist*, 2015.
[23] Henrieta Palubová. Chaotic sequences in MC-CDMA systems.
[24] Dixiong Yang, Zhenjun Liu, and Jilei Zhou. Chaos optimization algorithms based on chaotic maps with different probability distribution and search speed for global optimization. *Communications in Nonlinear Science and Numerical Simulation*, 19(4):1229–1246, 2014. https://doi.org/10.1016/j.cnsns.2013.08.017.
[25] S Talatahari, B Farahmand Azar, R Sheikholeslami, and AH Gandomi. Imperialist competitive algorithm combined with chaos for global optimization. *Communications in Nonlinear Science and Numerical Simulation*, 17(3):1312–1319, 2012. https://doi.org/10.1016/j.cnsns.2011.08.021.
[26] A Rezaee Jordehi. Chaotic bat swarm optimization (CBSO). *Applied Soft Computing*, 26:523–530, 2015. https://doi.org/10.1016/j.asoc.2014.10.010.
[27] Amir H Gandomi and Xin-She Yang. Chaotic bat algorithm. *Journal of Computational Science*, 5(2):224–232, 2014. https://doi.org/10.1016/j.jocs.2013.10.002.
[28] Robert M May. Simple mathematical models with very complicated dynamics. *Nature*, 261(5560):459–467, 1976. https://doi.org/10.1038/261459a0.
[29] Andy Reynolds. Liberating lévy walk research from the shackles of optimal foraging. *Physics of life reviews*, 14:59–83, 2015. https://doi.org/10.1016/j.plrev.2015.03.002.
[30] Arild O Gautestad and Atle Mysterud. The lévy flight foraging hypothesis: forgetting about memory may lead to false verification of Brownian motion. *Movement ecology*, 1(1):9, 2013. https://doi.org/10.1186/2051-3933-1-9.
[31] Nasif Shawkat, Shariba Islam Tusiy, and Md Arman Ahmed. Advanced cuckoo search algorithm for optimization problem. *International Journal of Computer Applications*, 132(2), 2015.
[32] Ling Ai Wong, Hussain Shareef, Azah Mohamed, and Ahmad Asrul Ibrahim. Optimal battery sizing in photovoltaic based distributed generation using enhanced opposition-based firefly algorithm for voltage rise mitigation. *The Scientific World Journal*, 2014, 2014.