

Improved African Buffalo Optimization Algorithm using Chaotic and Chaotic-Levy flight

Chinwe Peace Igiri ¹ · Yudhveer Singh ¹ · Deepshikha Bhargava

Received: date / Accepted: date

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 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 [3], Jacobian optimization method [2, 7], 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) [15], ABO [11] migrating bird optimization[30], etc. etc. have been proven to be more effective than the 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 [26].

The ABO algorithm, developed by Odili and Kahar [13], belongs to the class of metaheuristic population-based nature-inspired algorithms. ABO like Bat algorithm [28] employs the sense of sound, which varies from the breeding parasitism of cuckoo search [27], flashing and attraction of firefly [25], to the flocking characteristics of particle swarm optimization (PSO). Odili et al. [13] claimed that ABO was developed to address the problem

Chinwe Peace Igiri

¹Amity Institute of Information Technology, Amity University Rajasthan, India

Tel.: +917073837528

Fax: +

E-mail: chynkendirim@gmail.com

of premature convergence as well as balance exploration and exploitation by utilizing few parameters. However, this claim is not justifiable in all optimization problems. In terms of the application domain, ABO has been utilized to solve the traveling salesman problem (TSP) [13]. According to that study, ABO outperformed genetic algorithm (GA), ACO, honey-bees mating optimization, simulated annealing (SA), etc. [13]. The algorithm has also been tested on numerical functions with better results than GA and improved GA [12]. However, it was also found that chaotic gray-coded GA yielded better result than ABO [12]. In fact, that was the motivation to improve ABO using chaotic optimization. Further, ABO has shown better performance than randomization insertion algorithm in asymmetric TSP [14]. Additionally, it has outperformed PSO, GA, ACO, and bacterial foraging optimization in turning PID controller parameters. Recently, ABO has been utilized to solve budget constrained problem [7].

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 [26]. 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] developed an improved hybrid (ACO and GA) algorithm for industrial production operation. Also, Wang et al. [21], designed a hybrid GA and DE for joint replenishment and location inventory problem in a three-level supply chain. Li et al. [9] formulated hybrid genetic-simulated annealing algorithm (HGSAA) for supply chain environment. HGSAA outperformed GA in terms of computing time, optimal solution, and computing stability. Yu et al. [29] also formulated an adaptive hybrid of PSO and DE for global optimization in scientific and engineering fields. Furthermore, Duan et al. [4] 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.

2 Basic ABO

ABO, belonging to the class of swarm intelligence, was developed by Odili et al. [13]. 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 [13]. They utilize “waaa” and “maaa” sound to communicate danger and safety, respectively [13]. Thus, their organizational lifestyle could be mapped to these unique characteristics [22, 13]. The “waaa” sound is denoted by w_i , the “maaa” denoted by m_i , while learning parameter are denoted by l_1 and l_2 . Other parameters are global maximum (b_{ogmax}), the personal maximum (b_{opmax}) positions. The basic ABO is controlled by two equations, namely democratic Eq.1 and location update Eq.2 equations. Algorithm 2 shows the basic ABO. The algorithm subtracts the “waaa” value (w_i) asking the animals to explore the search space from the maximum vector (b_{ogmax} and b_{opmax}) which is further multiplied by the learning parameters (l_1 and l_2) [13]. The result is supplied by the “maaa” (m_i) value, this indicates that they herds should remain in that location and continue grazing.

$$m_{i+1} = m_i + l_1(b_{ogmax} - w_i) + l_2(b_{opmax} - w_i) \quad (1)$$

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

Algorithm 1 ABO [13]

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

2.1 Strengths and Weaknesses of ABO

From the foregoing, one could say that ABO has shown good solutions with better speed in TSP, benchmark numerical functions, etc. [13,14,12]. Also, the strength of ABO has been proven by low relative percentage error obtained in multi-modal and uni-modal functions [12]. The speed of this algorithm is due to the few parameters that control the algorithm. However, since there is no best algorithm for all optimization problems [26], ABO, like other basic metaheuristic algorithms, exhibits a number of weaknesses such as being inefficient for multi-variable optimization problems [12] 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 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 [13] 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 [26]. 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.

3 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. 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.

3.1 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 [16]. Many stochastic optimization problems are trapped in local optima. However, the literature shows that chaotic sequence or map could be used to deal with local optima [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 [20,8]. Basically, chaotic optimization is the use of a chaotic sequence to replace random variables in an optimization algorithm.

There are variations of one-dimensional chaotic maps including, logistic, iterative, sinusoidal, sin, circle, chebyshev, intermittency, singer, sawtooth, piecewise, tent, and liebovtch [5]. However, this study considers logistic map for ABO randomization. Eq.3 represents the logistic function [10,8].

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

3.2 Levy Flight

Levy walk is a description of a diffusion pattern of organisms such that searching is concentrated at the location of the potential solution. [18]. Levy flight foraging hypothesis estimates the migration from less-resource to more-resource environment, which results in optimal search [6]. Animals with high memory capability employ this model to explore their search space [6]. 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 be mathematically represented as

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

where $\psi > 0$ is a minimum step and ω is a scale parameter. Ideally as $\zeta \rightarrow \infty$, then

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

Mantegna algorithm [26] 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 [26]. In other words,

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

where

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

Readers are referred to [26] for details on Levy flight.

3.3 Proposed Improved ABO Algorithms

The improvement of ABO can be considered in two ways, namely Chaotic ABO (CABO) and Chaotic-Levy Flight ABO (CLABO).

3.3.1 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

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

we have the modified democratic and exploitation equation

$$m_{i+1} = m_i + l_1 \mu .* (b_{ogmax} - w_i) + l_2 \mu .* (b_{opmax} \cdot i - w_i) \quad (9)$$

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

where $.*$ is element-wise multiplication of vectors. The step by step implementation is shown in Algorithm 3.3.1

Algorithm 2 CABO

- Step1. Randomly initialize buffaloes within the search area
 Step2. Generate chaotic sequence or map
 Step3. Update buffaloes' exploitation with equation Eq. 10 enhanced by chaotic map
 Step4. Update the location of buffaloes with Eq. 9
 Step5. If equation 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
-

3.3.2 CLABO

Here, the aim is to locate the greenest pasture based on levy foraging hypothesis [6]. This would result in modification of both the democratic and location update equations as shown in Eqs. 11 and 12.

$$m_{i+1} = m_i + l_1 \mu .* (b_{ogmax} - w_i) + l_2 \mu .* (b_{opmax} \cdot i - w_i) \quad (11)$$

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

where $.*$ is element-wise multiplication of vectors, μ is the chaotic map and $L(\zeta)$ is the Levy flight. Algorithm 3.3.2 shows the step-wise implementation of the improved CLABO.

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
-

4 Results and Discussion

4.1 Numerical Experiments

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

1. Sphere (F1),

$$\varphi(x) = \sum_{i=1}^n x_i^2, x \in [-10, 10], 0 \quad (13)$$

2. Matyas (F2),

$$\varphi(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2, x \in [-10, 10], 0 \quad (14)$$

3. Six Hump Camel Back (F3),

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

4. Easom (F4),

$$\varphi(x) = -\cos(x_1)\cos(x_2)\exp\left(-\left(x_1 - \pi\right)^2 - \left(x_2 - \pi\right)^2\right), x \in [-100, 100], -1 \quad (16)$$

5. Ackley (F5),

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

6. Zakharov (F6),

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

7. Schaffer (F7),

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

8. Bochachvesky 1(F8),

$$\varphi(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7, x \in [-100, 100], 0 \quad (20)$$

9. Griewank (F9),

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

10. Rastrigin (F10) functions

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

The choice of these benchmark functions capture polynomials, transcendental, multidimensional and multimodal 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) [27], ICS [19], gravitational search algorithm (GSA) [23], enhanced opposition-based firefly algorithm (EOFA) [23], FA [23], and parallel migrating GA PMGA [23].

4.2 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 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 1 and 2. The ranking was used to summarize the comparative performance evaluation, as shown in table 3. Also, the mean performance was used as a comparative standard since it is

Table 1 Algorithms Performance Evaluation using Benchmark Functions

Functions	Algorithms	Iteration No.	Best	Worst	Mean	STD
F1	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
	ABO	100	2.84E-13	3.28E-11	3.28E-12	8.49E-12
	ICS[19]	1000	N/A	N/A	1.85	1.85
	ACS[19]	1000	N/A	N/A	4.71	5.87
F2	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
	ABO	100	2.29E-14	3.73E-12	5.17E-13	8.08E-13
	GSA [23]	1000	4.73E-09	9.11E-07	1.35E-07	N/A
	EOFA [23]	1000	1.59E-40	8.06E-36	1.45E-36	N/A
	FA [23]	1000	1.30E-05	0.58	0.04	N/A
F3	CLABO	100	-1.0316	-1.0249	-1.0300	0.0017
	CABO	100	-1.0316	-1.0301	-1.0311	0.0004
	ABO	100	-1.0316	-1.0311	-1.0315	0.0002
	ICS[19]	1000	N/A	N/A	3.3630	4.1773
	ACS[19]	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[19]	1000	N/A	N/A	1.7999	1.8598
	ACS[19]	1000	N/A	N/A	1.7495	1.7534
F5	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
	ABO	100	1.32E-06	7.53E-06	1.80E-06	1.44E-06
	ICS[19]	1000	N/A	N/A	3.6053	4.5822
	ACS[19]	1000	N/A	N/A	0.0288	2.3097
F6	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
	ABO	100	1.68E-11	2.75E-08	2.56E-08	2.56E-09
	GSA [23]	1000	6.80E-35	73.96	56.79	N/A
	EOFA [23]	1000	5.84E-35	1.60E-29	1.92E-30	N/A
	FA [23]	1000	708.22	3.92E+09	5.47E+08	N/A

the common statistical value available for all other algorithms. The two parameters, l_1 and l_2 that control the ABO algorithm are both set to 0.8 across both basic and proposed improved variants.

As mentioned earlier, the performances of the algorithms on the benchmark function are represented in tables 1 and 2. The best algorithm is highlighted in 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 3. Moreover, CABO and CLABO converged at global minimum for function F7 to F10 with relatively few iterations as shown in table 2. 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 a negative impact on these test functions. Additionally, only two convergence graphs, F3 1 and F7 2 have been selected for visual perception and at the same time ensure a simplified report.

5 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

Table 2 Algorithms Performance Evaluation using Benchmark Functions contd.

Functions	Algorithms	Iteration No.			Objective Value		
		Best	Worst	Mean	Best	Worst	Mean
F7	CLABO	42	50	46	0	0	0
	CABO	41	67	44	0	0	0
	ABO	100			4.91E-11	4.37E-02	2.91E-03
	PMGA [23]	10,000			N/A		12.63
	PSO [27]	10,000			N/A		5.30E-08
	FA [25]	10,000			N/A		6.63E-15
F8	CLABO	42	46	44	0	0	0
	CABO	41	67	44	0	0	0
	ABO	100			1.06E-11	6.22E-08	7.38E-09
	GSA [23]	1000			1.09E-06	1.05E-04	2.06E-05
	EOFA [23]	1000			0	2.22E-16	4.88E-17
	FA [23]	1000			2.10E-04	3.35	0.55
F9	CLABO	38	56	43	0	0	0
	CABO	35	72	48	0	0	0
	ABO	100			1.48E-12	0.05	0.01
	GSA [23]	1000			6.8E-06	0.03	1.78E-03
	EOFA [23]	1000			0	5.55E-17	1.78E-17
	FA [23]	1000			446.74	686.12	592.46
F10	CLABO	42	60	50	0	0	0
	CABO	42	65	49	0	0	0
	ABO	100			1.85E-10	1.0839	4.52E-02
	GSA [23]	1000			15.99	53.79	34.5
	EOFA [23]	1000			0	3.19	0.99
	FA [23]	1000			353.45	429.4	394.46

Table 3 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[19]	7	-	6	6	6	-	-	-	-	-	0.625	7
ACS[19]	6	-	7	7	7	-	-	-	-	-	0.675	6
EOFA [23]	-	8	-	-	-	8	-	7	8	7	0.76	4
GSA [23]	-	6	-	-	-	6	-	6	7	6	0.62	8
FA [23]	-	5	-	-	-	5	-	5	5	-	0.5	10
PMGA [23]	-	-	-	-	-	-	5	-	-	-	0.5	10
PSO [27]	-	-	-	-	-	-	6	-	-	-	0.6	9
FA [25]	-	-	-	-	-	-	7	-	-	-	0.7	5

n = number of benchmark functions

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 function 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 as illustrated in the tabulated results. Nevertheless, this study could be extended to constrained as well as real-world optimization problems such as process scheduling, engineering design, supply chain management, and vehicle routing, among others.

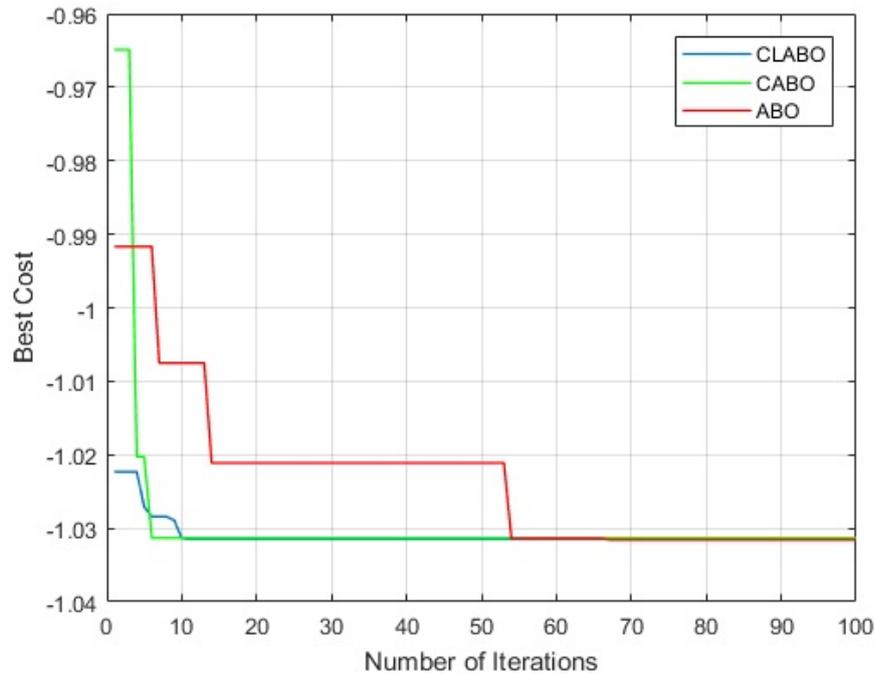


Fig. 1 Illustration of convergence of Six Hump Camel Back function

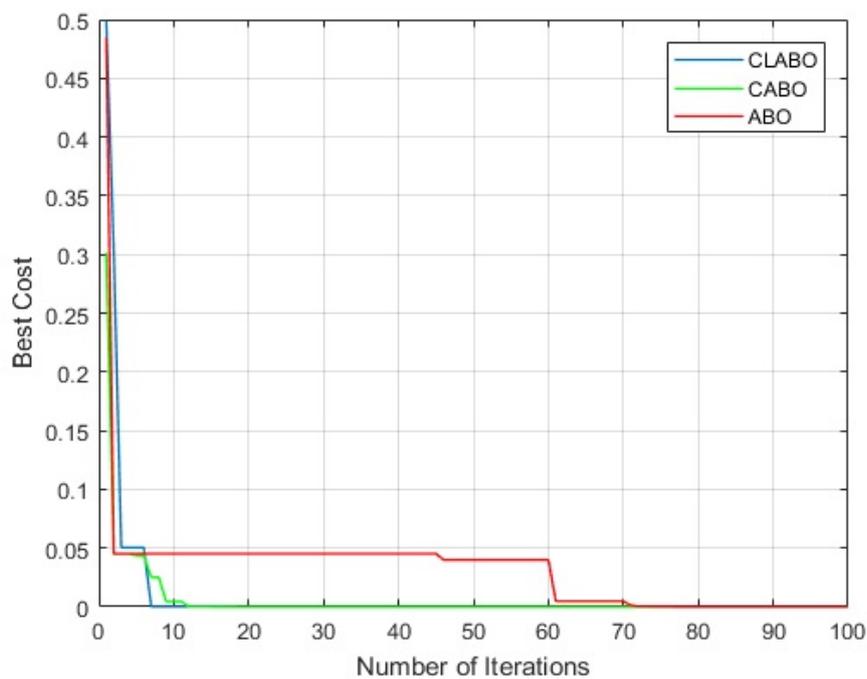


Fig. 2 Illustration of convergence of Schaffer function

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.
2. Vahid Beiranvand, Warren Hare, and Yves Lucet. Best practices for comparing optimization algorithms. *Optimization and Engineering*, 18(4):815–848, 2017.
3. EG Birgin, G Haeser, and A Ramos. Augmented lagrangians with constrained subproblems and convergence to second-order stationary points. *Optimization Online*, 2016.
4. 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.
5. Amir H Gandomi and Xin-She Yang. Chaotic bat algorithm. *Journal of Computational Science*, 5(2):224–232, 2014.

6. 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.
7. 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.
8. A Rezaee Jordehi. Chaotic bat swarm optimisation (cbso). *Applied Soft Computing*, 26:523–530, 2015.
9. 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.
10. Robert M May. Simple mathematical models with very complicated dynamics. *Nature*, 261(5560):459–467, 1976.
11. Julius Beneoluchi Odili, Mohammad Nizam Kahar, and A Noraziah. Solving traveling salesman?s problem using african buffalo optimization, honey bee mating optimization & lin-kerninghan algorithms. *World Applied Sciences Journal*, 34(7):911–916, 2016.
12. 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.
13. Julius Beneoluchi Odili, Mohd Nizam Mohmad Kahar, and Shahid Anwar. African buffalo optimization: a swarm-intelligence technique. *Procedia Computer Science*, 76:443–448, 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 salesman's problem. In *Software Engineering and Computer Systems (ICSECS), 2015 4th International Conference on*, pages 90–95. IEEE, 2015.
15. Francesco Orcioli, Mimmo Parente, and Autilia Vitiello. Solving the shopping plan problem through bio-inspired approaches. *Soft Computing*, 20(5):2077–2089, 2016.
16. Henrieta Palubová. Chaotic sequences in mc-cdma systems.
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. Andy Reynolds. Liberating lévy walk research from the shackles of optimal foraging. *Physics of life reviews*, 14:59–83, 2015.
19. Nasif Shawkat, Shariba Islam Tusiyy, and Md Arman Ahmed. Advanced cuckoo search algorithm for optimization problem. *International Journal of Computer Applications*, 132(2), 2015.
20. 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.
21. 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.
22. David Sloan Wilson. Altruism and organism: Disentangling the themes of multilevel selection theory. *The American Naturalist*, 2015.
23. 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.
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.
25. Xin-She Yang. Firefly algorithms for multimodal optimization. In *International symposium on stochastic algorithms*, pages 169–178. Springer, 2009.
26. Xin-She Yang. *Nature-inspired optimization algorithms*. Elsevier, 2014.
27. 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.
28. Xin-She Yang and Amir Hossein Gandomi. Bat algorithm: a novel approach for global engineering optimization. *Engineering Computations*, 29(5):464–483, 2012.
29. 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.
30. Biao Zhang, Quan-ke Pan, Liang Gao, Xin-li Zhang, Hong-yan Sang, and Jun-qing Li. An effective modified migrating birds optimization for hybrid flowshop scheduling problem with lot streaming. *Applied Soft Computing*, 52:14–27, 2017.