Article

Advanced Science, Engineering and Medicine
Vol. 8, 1–7, 2016
www.aspbs.com/asem

An Improved Butterfly Optimization Algorithm for Global Optimization

Sankalap Arora[1,*] and Satvir Singh[2]

[1]Research Scholar, I. K. Gujral Punjab Technical University, Jalandhar, Punjab, India
[2]Shaheed Bhagat Singh State Technical Campus, Ferozpur, Punjab, India

Butterfly optimization algorithm is a newcomer in the category of nature inspired optimization algorithm, which is inspired by the food foraging behavior of butterflies. It has shown remarkable performance on various global optimization problems. One of the important parameters which critically affect its performance is sensory modality, which accounts for sensing the different amount of fragrances emitted by butterflies of the search space and guide the butterflies to move towards the direction of the selected fragrance/butterfly. However, in the basic butterfly optimization algorithm, fixed value of sensory modality is adopted throughout all generations. This results in two problems, (1) the algorithm will easily get trapped in local optima and (2) low accuracy. In order to solve these problems, an improved butterfly optimization algorithm is proposed which employs variable sensory modality parameter strategy. Various benchmark functions are used to validate the performance of the proposed algorithm. Its performance is compared with the basic butterfly optimization algorithm and three other population-based optimization algorithms viz. artificial bee colony, firefly algorithm, and particle swarm optimization. The simulation results demonstrate that the proposed algorithm performs better, or at least comparable, in terms of the final solution quality and convergence rate.

Keywords: Butterfly Optimization Algorithm, Variable Sensory Modality Parameter, Global Optimization, Local Optima.

1. INTRODUCTION

Real world problems are very complex in nature and require optimal solution in less computational time. However, traditional methods failed to provide efficient results in shorter amount of time.[1] To overcome the limitations of traditional methods, nature inspired optimization algorithms came into existence.[2] Nature inspired optimization algorithms have received much attention by various researchers in the past.[3] Their potential has recognized themselves as global optimization techniques in various real world complex problems.[4] These algorithms find their source of inspiration in nature. Various algorithms have been proposed in the past like Particle Swarm Optimization (PSO),[5] Firefly Algorithm (FA),[6] Artificial Bee Colony (ABC)[7] and many more.[8,9] These algorithms demonstrate improved performances when compared with conventional optimization techniques, especially when applied to solve non-convex optimization problems.

Butterfly Optimization Algorithm (BOA) is a nature inspired optimization algorithm used for solving global optimization problems.[10] The underlying mechanism of BOA is to mimic the food searching abilities of biological butterflies. Preliminary studies suggest that the BOA demonstrate superior results, when compared with other population based optimization algorithms.[11] However, BOA has tendency to show premature converge to local optima. In order to improve the performance of BOA, recently chaos is introduced in BOA so as to increase the global search mobility for global optimization problems.[12] As a matter of fact, in the basic BOA, the sensory modality parameter which distinguishes different amount of fragrances emitted by the butterflies is constant throughout all the generations. This is undoubtedly wrong and it may affect the proper balance of global and local search. The sensory modality parameter must be adjusted adaptively and dynamically in order to set appropriate value of sensory modality for each generation. So in this study, a variable sensory modality parameter strategy is employed in BOA. Moreover, in the basic butterfly optimization

*Author to whom correspondence should be addressed.

algorithm, the randomness is employed using lèvy flights whereas in the proposed algorithm pseudo-random numbers are used.

The rest of this paper is structured below. To start with, a brief background on the BOA is provided in Section 2. Our proposed BOA with variable sensory modality parameter strategy is presented in Section 3. Different benchmark functions which are used in this study are described in Section 4, and finally, conclusions are drawn in Section 5.

2. BUTTERFLY OPTIMIZATION ALGORITHM

Nature inspired optimization algorithms have gained much attention by various researchers in the past.[3] Butterfly Optimization Algorithm (BOA) belongs to the same class of optimization algorithms. It is basically inspired by the food foraging behavior of butterflies. In BOA, these butterflies are used as search agents in order to perform optimization.[11] Biologically, butterflies have sense receptors, which are used to smell/sense fragrance of their food/flowers. These sense receptors are called chemoreceptors which are scattered over the butterfly's body parts like legs, palps, antennae etc.[13] In BOA, it is assumed that each butterfly is able to generate fragrance with some intensity. This fragrance is further correlated with fitness of the butterfly. This means that whenever a butterfly moves from one position to other particular position in the search space, its fitness will vary accordingly. Now, the fragrance which is generated by butterflies is propagated over distance to all other butterflies in the region. The propagated fragrance is sensed by other butterflies and a collective social knowledge network is formed. Whenever a butterfly senses fragrance from the best butterfly in the search space, it takes a step towards the best butterfly and this phase is termed as global search phase of BOA. In the another scenario, when a butterfly is not able to sense fragrance of any other butterfly in the search space, it will take random steps and this phase is termed as local search phase in BOA.

The main strength of BOA lies in its mechanism to modulate fragrance in the algorithm. In order to understand the modulation, first it should be discussed that how any sense like sound, smell, heat, light etc. is processed by a stimulus of a living organism. The basic concept of sensing is dependent on three vital parameters i.e., sensory modality (c), stimulus intensity (I) and power exponent (a). Sensory modality defines the method by which the form of energy is measured and processed by the sensors. Different modalities/senses can be smell, sound, light, temperature or pressure etc. and in BOA, it is fragrance. In BOA, many butterflies emit fragrance at the same time, it is their sensory modality which allows butterflies to sense and differentiate these fragrances from each other. I represents the magnitude of the physical/actual stimulus and in BOA, it is correlated with the fitness of the butterfly/solution i.e., a butterfly with higher fragrance or greater fitness value attracts other butterflies in that region. The parameter a allows response compression i.e., as the stimulus gets stronger, insects become increasingly less sensitive to the stimulus changes.[14,15] Summarizing, the natural phenomenon of butterflies is based on two imperative issues: (1) variation of I, (2) formulation of f. For simplicity, in BOA, I of a butterfly is associated with encoded objective function. However, f is relative i.e., it should be sensed by other butterflies. Therefore, considering these concepts, in BOA, the fragrance is formulated as a function of the physical intensity of stimulus[15] as follows:

$$f_i = cI^a \tag{1}$$

where f_i is the perceived magnitude of fragrance, i.e., how stronger the fragrance is perceived by ith butterfly, c is the sensory modality, I is the stimulus intensity and a is the power exponent dependent on modality, which accounts degree of absorption. There are two important phases in the BOA, they are; global search phase and local search phase. In global search phase, the butterfly takes a step towards the fittest butterfly/solution g^* which can be represented as:

$$x_i^{t+1} = x_i^t + (l\grave{e}vy(\lambda) \times g^* - x_i^t) \times f_i \tag{2}$$

where x_i^t is the solution vector x_i for ith butterfly in iteration t. Here g^* represents the best solution found among all the solutions in current generation. The fragrance of ith butterfly is represented by f_i while step size is represented by λ.

Algorithm 1 (Buttery Optimization Algorithm).

```
Objective function f(x), x = (x_1 ... x_d)
where d is number of dimensions.
Generate initial population of butterflies
x_i = (i = 1, 2, ... n)
Find the best solutions g* in the initial population
Define switch probability p ∈ [0, 1]
while stopping criteria not met do
    for each butterfly in population do
        Draw rand from a uniform distribution
        in [0, 1]
        Calculate fragrance of the butterfly
        if rand < p then
            Global search using Eq. (2)
        else
            Randomly choose j and k
            among all the solutions
            Do Local search using Eq. (3)
        end if
        Evaluate new solutions
        If new solutions are better, update them
        in population
    end for
    Find the current best solution g*
end while
Output the best solution found.
```

Local search phase can be represented as:

$$x_i^{t+1} = x_i^t + (\text{l\`evy}(\lambda) \times x_k^t - x_j^t) \times f_i \quad (3)$$

where x_j^t and x_k^t are jth and kth butterflies from the solution space. If x_j and x_k belongs to the same sub-swarm and λ is the step size, then Eq. (3) becomes a local random walk. The food searching process can occur at local as well as global level, so considering this, a switch probability p is used in BOA to control the common global search and intensive local search. The above mentioned steps frame the complete algorithm of Butterfly Optimization Algorithm and its pseudocode is presented in Algorithm 1.

3. THE PROPOSED BUTTERFLY OPTIMIZATION ALGORITHM

3.1. Analysis of Basic Butterfly Optimization Algorithm

Every nature inspired optimization algorithm, must balance the trade-off between global and local search, as it is very crucial in finding the optima efficiently. During the early stages of optimization, it is always desired that the solutions are encouraged to wander through the entire search space, without gathering around the local optima. Whereas in order to find the optimum solution, it is essential to converge towards the global optimum solution in the later stages of optimization.[16]

Sensory modality c is one of the important parameters in the basic butterfly optimization algorithm. The importance of c can be judged from the fact that it enables each butterfly in the search process to sense the fragrances emitted by other butterflies and guide the search towards those butterflies. This means that better the sensing mechanism, more efficient the results will be. However, the static method of setting c will not be adaptive to complex real world problems. It will effect the performance of algorithm in two ways, firstly, if a large value of c is used then it may skip the best optimal solution in the early stage of optimization which will reduce the search performance of the algorithm. Secondly, if a small value of c is used then it may fall into local optima trap problem and will result in premature convergence.

Therefore, the sensory modality has a great impact on the searching ability of butterflies. The value of c should be increase rapidly with small the number of generations while it should increase slowly with the large number of generations. This will definitely increase the effectiveness of the algorithm. Considering the above problems and importance of sensory modality parameter, the algorithm is modified in such a manner that the butterflies are able to change the value of c dynamically.

3.2. Variable Sensory Modality Parameter of Butterfly Optimization Algorithm

By using static value of sensory modality c the searching process of butterfly optimization algorithm is not used efficiently. Theoretically, a large value of c will enable the butterflies to explore new search space, however, it will have adverse effect on the convergence towards global optimum solution. Whereas if a small value of c is used, the results will be perverse. This means c has great effect on the searching abilities of the butterflies and if the value of c is modified according to the requirements of stage of optimization process, it will prove beneficial to the performance of the algorithm. So in this study, a dynamic and adaptive adjusting strategy of sensory modality is designed and used. The sensory modality c can be calculated as following:

$$c^{t+1} = c^t + \left(0.025/(c^t \times MaxIterations)\right) \quad (4)$$

where t is the current number of iterations and *MaxIterations* is the maximum number of iterations. The increasing nature of dynamic parameter strategy is graphically illustrated in Figure 1. It can be analyzed from this figure that the value of c is small initially and then it increases with the increase in number of iterations. It will help the butterflies to improve their search abilities. Another modification which is proposed in the basic butterfly algorithm is that in Eqs. (2) and (3), lèvy flights are used in which the step-lengths have a probability distribution that is heavy-tailed. In this study, pseudorandom numbers are used instead of lèvy flights. To take into account the above discussions, the global search and local search phase of proposed Improved Butterfly Optimization Algorithm (IBOA) are described in Eqs. (5) and (6) respectively.

$$x_i^{t+1} = x_i^t + (r^2 \times g^* - x_i^t) \times f_i \quad (5)$$

where x_i^t is the solution vector x_i for ith butterfly in iteration number t. Here g^* represents the current best solution found among all the solutions in current stage. Fragrance of ith butterfly is represented by f_i and r is a random number in [0, 1].

Local search phase can be represented as:

$$x_i^{t+1} = x_i^t + (r^2 \times x_k^t - x_j^t) \times f_i \quad (6)$$

where x_j^t and x_k^t are jth and kth butterflies from the solution space and r is a random number in [0, 1].

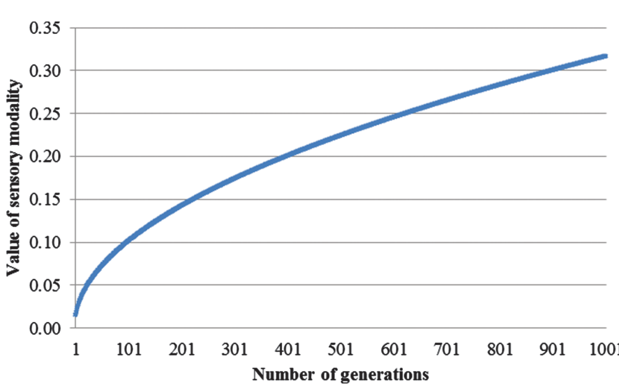

Figure 1. Variable sensory modality.

4. SIMULATION RESULTS AND ANALYSIS

There are various benchmark functions which are used by researchers to validate new optimization algorithms.[17] However, no standard set of functions is defined but still different benchmark functions are recommended by various researchers in the past.[18] These benchmark functions are categorized on basic of modality, dimensionality and separability. In this paper, a diverse subset of benchmark functions is chosen in order to validate the proposed algorithm, as shown in Table I.[19] In the first category, unimodal functions, i.e., $f_1 - f_6$, which tests the convergence speed of the algorithm are used. Then, the most difficult category of functions i.e., multimodal functions, f_7–f_{15}, which test the algorithm's capability to find global optima when the number of local optima increases exponentially with problem dimension. Further in the third category of dimensionality, the algorithm is validated on low dimensional functions i.e., f_1, f_2 and f_{11}–f_{15}, and high dimensional functions i.e., f_3–f_{10}. In the last category of separability, the separable functions which are used in this study are f_3–f_5, f_7, f_8, f_{10}, f_{14} and f_{15}. In addition to these categories, step function i.e., f_3 which is discontinuous and has only one minima is used. Special attention should be paid to the noisy (Quartic) function i.e., f_4, as these challenges occur frequently in real-world applications.

4.1. Experimental Setup

All the experiments are conducted under the same conditions in order to obtain fair results as shown in Ref. [20]. In this study, basic version of ABC is used which employs only one control parameter $limit$ whose value is calculated by $limit = SN \times D$ where D is the dimension of the problem and SN is the number of food sources or employed bees. For PSO, we used only global learning (no local neighborhoods), an inertial constant = 0.3, a cognitive constant = 1, and a social constant for swarm interaction = 1. For FA, $\alpha = 0.2$, $\beta_0 = 1$ and $\gamma = 1$ are used. For the proposed IBOA and BOA, the value of sensory modality c and power exponent a is set to 0.01 and 0.1 respectively.[11] It is worth mentioning here that in IBOA and in basic BOA, initially the value of sensory modality c is set to 0.01. In IBOA it is varied according to the Eq. (4), whereas in basic BOA is fixed. These parameters are set as reported by the authors in the past.[7, 11, 21, 22]

Rigorous nonparametric statistical framework is used to compare the performance of the proposed algorithm with other selected optimization algorithms. For each run of the algorithm, all initial solutions of the population are randomly generated. The population size n is fixed to 30 for all the algorithms. Dimensions and range of the benchmark functions is given in the Table I. The proposed IBOA is implemented in C++ and compiled using Qt Creator 2.4.1 (MinGW) under Microsoft Windows 8 operating system. All simulations are carried out on a computer with an Intel (R) Core (TM) i5-3210@2.50 Ghz CPU.

4.2. General Performance of IBOA

In order to better analyze the performance analysis of proposed IBOA against the basic BOA, simulation results of

Table I. Benchmark functions used in current experimental study.

S. no.	Benchmark function	Formula	Dim	Range	Optima
f_1	Booth	$f(x) = (x_0 + 2_{x1} - 7)^2 + (2x_0 + x_1 - 5)^2$	2	$(-10, 10)$	0
f_2	Matyas	$f(x) = 0.25(x_0^2 + x_1^2) - 0.48x_0x_1$	2	$(-10, 10)$	0
f_3	Step	$f(x) = \sum_{i=0}^{n-1} (\lfloor x_i \rfloor + 0.5)^2$	30	$(-100, 100)$	0
f_4	Quartic function with noise	$f(x) = \sum_{i=0}^{n-1} x_i^4 + N(0, 1)$	30	$(-1.281.28)$	0
f_5	Schwefel 2.21	$f(x) = \max(\lvert x_i \rvert)$	30	$(-10, 10)$	0
f_6	Rosenbrock	$f(x) = \sum_{i=1}^{n-1} 100(x_i - x_{i-1}^2)^2 + (1 - x_{i-1})^2$	30	$(-10, 10)$	0
f_7	Sphere	$f(x) = \sum_{i=1}^{n} x_i^2$	30	$(-100, 100)$	0
f_8	Levy	$f(x) = \sin^2(3\Omega x_0) + \sum_{i=0}^{n-2} (x_i - 1)^2(1 + \sin^2(3\Omega x_{i+1})) + (x_{n-1} - 1)(1 + \sin^2(2\Omega x_{n-1}))$	30	$(-10, 10)$	-21.5023
f_9	Griewank	$f(x) = \frac{1}{4000}\sum_{i=1}^{n-1} (x_i - 100)^2 - \prod_{i=1}^{n-1} \cos\left(\frac{x_i - 100}{\sqrt{i-1}}\right) + 1$	30	$(-600, +600)$	0
f_{10}	Schwefel 2.26	$f(x) = -\sum_{i=0}^{n-1} x_i \sin\sqrt{x_i}$	30	$(-500, 500)$	-12569.5
f_{11}	Power sum	$f(x) = \sum_{k=0}^{n-1}\left(\left[\sum_{i=0}^{n-1} x_i^{k+1}\right] - b_k\right)^2$	4	$(0, n)$	0
f_{12}	Shekel4.5	$f(x) = \sum_{i=0}^{m} \frac{1}{(x - A_i)^t(x - A_i) + c_i}$	4	$(0, 10)$	-10.1532
f_{13}	Bohachevsky	$f(x) = x_0^2 + 2x_1^2 - 0.3\cos(3\Omega x_1) - 0.4\cos(4\Omega x_1) + 0.7$	2	$(-100, 100)$	0
f_{14}	Shubert	$\left(\sum_{i=1}^{5} i\cos((i+1)x_0 + i\right)\left(\sum_{i=1}^{5} i\cos((i+1)x_1 + i\right)$	2	$(-10, 10)$	-186.73
f_{15}	Easom	$f(x) = -\cos(x_0)\cos(x_1)\exp(-(x_0 - \Omega)^2 - (x_1 - \Omega)^2$	2	$(-100, 100)$	-1

Table II. Simulation results of IBOA and BOA.

	BEST		Worst		Mean		Std. dev.	
	IBOA	BOA	IBOA	BOA	IBOA	BOA	IBOA	BOA
f_1	4.12E−06	1.32E−03	2.16E−03	7.04E−02	6.83E−04	6.04E−02	6.70E−04	2.12E−02
f_2	2.59E−17	3.81E−05	7.71E−15	6.10E−04	1.11E−15	1.74E−03	2.38E−15	2.10E−04
f_3	0.00E+00	5.10E+01	0.00E+00	7.40E+01	0.00E+00	1.26E+02	0.00E+00	6.92E+00
f_4	9.99E−04	6.73E−01	2.22E−03	8.01E−01	1.59E−03	7.43E+00	4.36E−04	4.08E−02
f_5	3.44E−10	2.02E−01	5.72E−10	2.28E−01	4.51E−10	2.15E+00	8.19E−11	9.01E−03
f_6	2.87E+01	2.88E+01	2.89E+01	2.89E+01	2.88E+01	9.59E+01	3.12E−02	4.13E−02
f_7	1.90E−114	6.33E−44	1.90E−103	4.35E−37	1.90E−104	9.07E−37	6.01E−104	1.63E−37
f_8	6.20E−01	7.71E−01	8.41E−01	1.93E+00	6.97E−01	9.93E+00	7.24E−02	3.90E−01
f_9	8.97E−13	8.70E−02	1.61E−12	1.01E−01	1.44E−12	9.41E−01	2.12E−13	4.55E−03
f_{10}	−3.03E+03	−3.23E+03	−1.39E+03	−1.50E+03	−1.77E+04	−1.57E+04	2.92E+02	4.90E+02
f_{11}	4.25E−02	2.02E−02	7.30E−01	4.81E−01	2.42E−01	3.45E−01	1.87E−01	1.47E−01
f_{12}	1.66E−16	2.46E−02	4.21E−13	4.57E−01	5.01E−14	3.21E+00	1.31E−13	1.89E−01
f_{13}	1.71E+00	1.72E+00	1.71E+00	1.72E+00	1.71E+00	1.72E+00	0.00E+00	0.00E+00
f_{14}	3.18E−12	4.13E−01	3.25E−11	7.06E−01	2.30E−11	5.46E+00	8.62E−12	7.72E−02
f_{15}	3.00E+00	3.57E+00	3.73E+00	3.06E+01	3.13E+00	9.36E+00	2.29E−01	8.70E+00

best solutions, worst solutions, mean value and standard deviations on IBOA and basic butterfly optimization algorithm are presented in Table II. The best values are highlighted in bold. The number of iterations is fixed to 500 in this comparative study. It can analyzed from the simulation results that the proposed approach is superior than the basic butterfly optimization algorithm in terms of precision and efficiency.

In order to fully evaluate the proposed algorithm, it is compared with other state-of-art approaches viz. Artificial Bee Colony (ABC), Firefly Algorithm (FA) and Particle Swarm Optimization (PSO). The reason behind the selection of these algorithms is their applicability to world wide applications. These algorithms have shown good performance in the past when applied to real world problems.[23]

Table III. Comparative performance of proposed algorithm with other approaches.

S. No.		ABC	BOA	IBOA	FA	PSO
f_1	Mean	2.07E+04	1.55E−01	1.99E−03	4.64E−10	4.68E−09
	Std. dev.	1.02E+04	1.21E−01	2.37E−03	2.50E−10	7.01E−09
f_2	Mean	1.86E+04	2.69E−03	5.95E−09	5.65E−12	1.84E−09
	Std. dev.	9.11E+03	1.28E−03	4.76E−09	3.46E−12	3.06E−09
f_3	Mean	6.31E+04	5.84E+02	0.00E+00	1.94E+03	1.04E+04
	Std. dev.	2.35E+04	1.03E+01	0.00E+00	8.57E+02	3.32E+03
f_4	Mean	5.66E+04	2.80E+01	3.64E−03	3.42E+05	2.11E+01
	Std. dev.	2.89E+04	2.49E+00	1.15E−03	2.26E+05	1.43E+00
f_5	Mean	2.63E+04	2.81E+00	8.25E−05	1.56E+00	7.73E+00
	Std. dev.	1.04E+04	8.24E−02	7.33E−06	4.86E−01	7.61E−01
f_6	Mean	3.59E+04	2.89E+02	2.89E+01	3.38E+03	6.32E+01
	Std. dev.	1.14E+04	5.06E+00	2.95E−02	1.91E+03	3.23E+01
f_7	Mean	6.97E+04	1.11E−04	1.03E−11	5.26E+10	1.26E+04
	Std. dev.	7.82E+03	1.15E−04	2.95E−11	3.34E+09	5.48E+03
f_8	Mean	3.73E+04	1.17E+01	8.06E−01	4.51E+01	7.90E+01
	Std. dev.	2.33E+04	7.80E−01	1.22E−01	1.60E+00	2.91E+01
f_9	Mean	4.49E+04	1.73E+01	1.16E−06	2.18E+01	8.49E+01
	Std. dev.	2.84E+04	6.69E−01	1.88E−07	8.09E+00	2.91E+01
f_{10}	Mean	2.33E+04	−1.27E+04	−2.37E+04	−3.89E+03	5.21E+01
	Std. dev.	1.07E+04	3.95E+02	4.13E+02	5.36E+02	1.70E+01
f_{11}	Mean	1.70E+04	2.00E+00	4.38E−01	1.15E+08	5.21E−01
	Std. dev.	7.67E+03	3.31E−01	1.81E+01	1.27E+08	4.37E−01
f_{12}	Mean	4.99E+04	4.65E+00	4.58E−07	1.79E−07	4.30E−06
	Std. dev.	2.88E+04	3.43E+00	6.17E−07	1.54E−08	8.10E−06
f_{13}	Mean	2.06E+01	1.72E+01	1.72E+00	9.10E+00	2.00E+01
	Std. dev.	1.10E−01	0.00E+00	0.00E+00	1.01E+00	7.25E−02
f_{14}	Mean	1.95E+11	9.81E+00	3.85E−05	1.77E+01	1.02E+02
	Std. dev.	4.12E+11	1.73E+00	3.31E−05	3.71E+00	3.14E+01
f_{15}	Mean	1.69E+01	5.31E+01	5.31E+00	5.70E+00	3.00E+00
	Std. dev.	2.12E+01	4.80E+00	8.75E+00	8.54E+00	0.00E+00

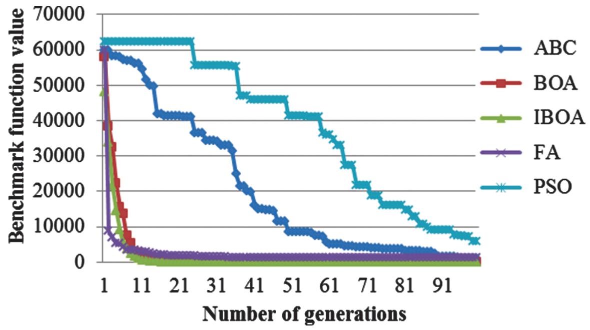

Figure 2. The convergence curves of f_3.

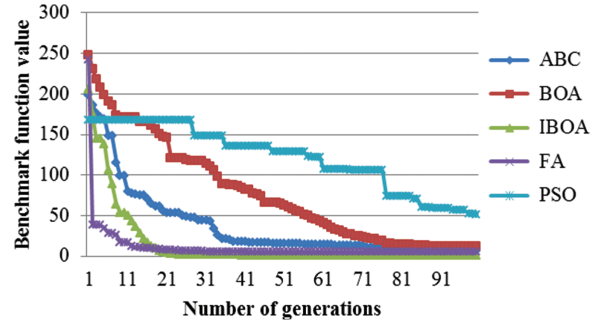

Figure 4. The convergence curves of f_8.

The number of iterations used for this particular comparative analysis is fixed to 100 for all the algorithms. From Table III, it can be observed that among all the approaches used, IBOA performs superior in eleven benchmark functions among the fifteen benchmark functions used in this study. Moreover, in the remaining four functions, IBOA demonstrate competitive results. It is worth pointing out that IBOA has shown better results than BOA on every benchmark function. This clearly shows the dominance of IBOA over BOA and other approaches.

Due to limitations of space, only three representative convergence graphs on benchmark functions are shown in Figures 2–4.

From Figures 2 and 3, it is analyzed that IBOA performs significantly better than BOA and other population based algorithms. By carefully looking at Figure 3, it can be analyzed that in the beginning of the optimization process FA converges faster than IBOA however in the later stages of optimization the IBOA is able to improve its solution quality, steadily. The reason might be that in IBOA, the varied value of sensory modality of the butterflies allowed the butterflies to find better solutions in the search space. The striking potential of proposed algorithm can be analyzed from the convergence curves shown in the Figure 4. The proposed algorithm's performance is also well in terms of convergence. FA shows faster convergence than IBOA in first ten iterations, however IBOA shows its superiority afterwards by improving the quality of solutions.

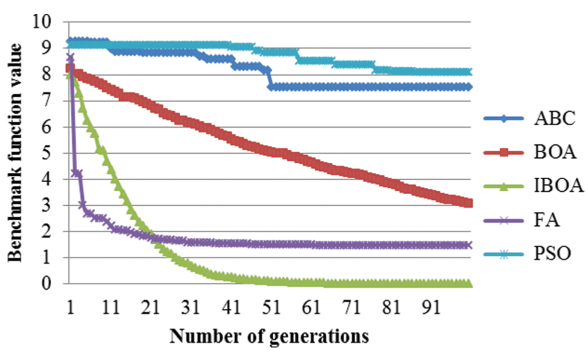

Figure 3. The convergence curves of f_5.

From the simulation results it can be analyzed that the variation of sensor modality proved to a significant improvement in the butterfly optimization algorithm. In the improved version of butterfly optimization algorithm, the butterflies are able to better explore the search space which lead them to find better solutions. The improvement has enhanced the performance of butterfly optimization algorithm in terms of convergence as well as solution quality.

5. CONCLUSION

In the present work, a variable sensor modality butterfly optimization algorithm is proposed for global optimization problems. The proposed algorithm uses a dynamic and adaptive strategy to modify the sensor modality which was fixed to a constant value in basic butterfly optimization algorithm. The variable value of sensor modality enhanced the searching abilities of the butterflies. Fifteen benchmark functions are used to investigate the performance of proposed algorithm against basic butterfly optimization algorithm and other population based optimization algorithms. The results demonstrated that in the proposed algorithm, the butterflies make effective use of their information to perform exploration and exploitation, in an efficient manner, than basic BOA. In the present study only unconstrained problems are considered, whereas it will be interesting to see the performance of improved butterfly optimization algorithm on constrained problems.

Acknowledgment: The authors wish to acknowledge the Department of RIC, I. K. Gujral Punjab Technical University, Kapurthala, Punjab, India.

References and Notes

1. J. S. Arora, Introduction to Optimum Design, Elsevier Academic Press **(2004)**.
2. I. Fister, Jr., X.-S. Yang, I. Fister, J. Brest, and D. Fister, *Elektrotehniski Vestnik* 80, 116 **(2013)**.
3. X.-S. Yang, Nature-Inspired Metaheuristic Algorithms, Luniver Press **(2010)**.
4. T. Back, Evolutionary Algorithms in Theory and Practice, Oxford Univ. Press **(1996)**.

5. J. Kennedy, Encyclopedia of Machine Learning, Springer **(2010)**, pp. 760–766.

6. X.-S. Yang and X. He, *International Journal of Swarm Intelligence* 1, 36 **(2013)**.

7. D. Karaboga and B. Basturk, *Journal of Global Optimization* 39, 459 **(2007)**.

8. G. C. Onwubolu and B. Babu, New Optimization Techniques in Engineering, Springer Science and Business Media **(2004)**, Vol. 141.

9. P. Vasant, Handbook of Research on Artificial Intelligence Techniques and Algorithms, Information Science Reference-Imprint of: IGI Publishing **(2015)**, Vol. 1–2.

10. S. Arora and S. Singh, *Proceedings of National Conference on Latest Initiatives and Innovations in Communication and Electronics (IICE 2015)*, pp. 69–72.

11. S. Arora and S. Singh, *Proceeding of the International Conference on Signal Processing, Computing and Control (2015 ISPCC)*, Himachal Pradesh, India, September, pp. 220–224.

12. S. Arora and S. Singh, *Journal of Intelligent and Fuzzy Systems* (in press).

13. R. B. Blair and A. E. Launer, *Biological Conservation* 80, 113 **(1997)**.

14. J. J. Zwislocki, Sensory Neuroscience: Four Laws of Psychophysics: Four Laws of Psychophysics, Springer Science and Business Media **(2009)**.

15. S. S. Stevens, Psychophysics, Transaction Publishers **(1975)**.

16. A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, *IEEE Transactions on Evolutionary Computation* 8, 240 **(2004)**.

17. M. Jamil and X.-S. Yang, *International Journal of Mathematical Modelling and Numerical Optimisation* 4, 150 **(2013)**.

18. D. Simon, *IEEE Transactions on Evolutionary Computation* 12, 702 **(2008)**.

19. J. Liang, B. Qu, P. Suganthan, and A. G. Hernandez-Daz, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report, 201212 **(2013)**.

20. G. Wang, L. Guo, H. Wang, H. Duan, L. Liu, and J. Li, *Neural Computing and Applications* 24, 853 **(2014)**.

21. R. C. Eberhart and Y. Shi, *Proceedings of the Congress on Evolutionary Computation*, Portland, US, June **(2001)**, Vol. 1.

22. X.-S. Yang, *Engineering Optimization* 221 **(2010)**.

23. J. Brownlee, Clever Algorithms: Nature-Inspired Programming Recipes, LuLu, Melbourne **(2011)**.