Improved Whale Optimization Algorithm for Numerical Optimization

A K Vamsi Krishna[1] and Tushar Tyagi[1]

[1] School of Electronics and Electrical Engineering , Lovely Professional University
krish.0724@gmail.com
[2] School of Electronics and Electrical Engineering, Lovely Professional University
tushar.20586@lpu.co.in

Abstract. In this paper an Improved Whale Optimization Algorithm which is intended to-wards the better optimization of the solutions under the category of meta-heuristic algorithms is proposed. Falling under the genre of Nature Inspired Algorithms, the Improved Whale Optimization delivers better results with comparatively better convergence techniques used. A detailed study and comparative analysis has been made between the principal and the modified algorithms and a variety of fit-ness functions has been used to confirm the efficiency of the Improved Algorithm over the older version. The merits with Nature Inspired Algorithms include distributed computing, reusable components, network processes, mutations and crossovers leading to better results, randomness and stochasticity.
Keywords: Optimization Algorithms, Nature Inspired Algorithms, Whale Optimization Algorithm, Bubble Net Forging, Meta heuristic algorithm, Randomization, Fitness Function, Spiral Updating, Encircling Mechanism

1 Introduction

The nature around us provides us with many efficient and optimal ways to solve problems. The Nature Inspired Algorithms are based on the Optimization techniques offered by the nature and in a sort of way, they imitate or mimic the processes available in nature where artificial intelligence and man's intellectual abilities fail. Nature Inspired Algorithms along with Fuzzy Logic systems and Neural Networks fall under the Category of Computational Intelligence. The use of Nature in problem solving and Optimizing real world problems can be attributed to the design and construction of Aero plane wings based on the Wings of Eagle, turbine blade design based on the swimming of Dolphins etc.[1]

1.1 Nature inspired Algorithms

The nature inspired algorithms can be basically divided into two concepts, one being evolutionary algorithms and the other being Swarm Optimization. Evolutionary Algorithms include Genetic algorithm, Differential Evaluation algorithms and many others where the Algorithm can be constantly improved with newer solutions available. [1][2]

Examples. A few of the Nature Inspired Algorithms are Firefly Algorithm, Migrating Birds Optimization, Ant Lion Optimization, etc.

2 Whale Optimization Algorithm

Whale Optimization algorithm abbreviated as WOA is an optimization algorithm based on the Hunting process or mechanism of hump back whales. This is a recently developed algorithm in the year 2016.It falls under the category of Meta heuristic algorithm and use a compromise between the local search and randomization. This is a better approach to move from a localized scale of search to global level search. WOA was proposed by Lewis and Mirjalili based on the simulation of bubble net method of Humpback whales during the hunt for their prey.[3][4]

2.1 About WOA

Introduced by Mirjalili in the year 2016, WOA is simulation based algorithm predominantly characterized depending on Hunting technique or Hunting Style or Humpback Whales. The interesting aspect of their hunting technique is that the whales create distinctive bubbles along a spiral or circular path as the whales continue to encircle the Krill or Prey. This distinct technique is referred to as ‘Bubble Net Foraging Method. The path traced by the bubbles can also be in the Number 9 shaped path or a simple circular path.[3-5]

2.2 Mathematical Modelling

This technique is forged into a mathematical model which consists of three aspects of encircling the prey, bubble net forging and search for the prey. .[3-5]

1) ENCIRCLING THE PREY :
Humpback whales first encircle the prey and then refresh and renew their position moving towards the best search from start to a maximum number of specified iterations given by equations (1) and (2). The number of iterations are specified during the algorithm and the more number of iterations generally means much closer to the optimal solution.[4]
Mathematically,

$$\vec{E} = \left| \overrightarrow{D} . \overrightarrow{Y^*}(t) - \overrightarrow{Y}(t) \right| \quad (1)$$

$$\overrightarrow{Y}(t+1) = \overrightarrow{Y^*}(t) - \vec{B}.\vec{E} \quad (2)$$

where,
$\vec{B}$ and $\overrightarrow{D}$ are coefficient vectors,
t indicates the present iteration,
Y^* is the position vector of the best solution obtained so far,
$\vec{Y}$ is the position vector,

|| is the absolute value and
. is an element-by-element multiplication
The vectors $\vec{B}$ and $\overrightarrow{D}$ are given by equations (3) and (4) as follows :

$$\vec{B} = 2\vec{b}.\vec{r} - \vec{b} \tag{3}$$

$$\overrightarrow{D} = 2.\vec{r} \tag{4}$$

where,
$\vec{b}$ is linearly decreased from 2 to 0 as iterations progress
$\vec{r}$ is a random vector in [0, 1].

2) BUBBLE NET ATTACKING METHOD :
For mathematical modelling of Bubble Net attack method, there are two mechanisms to be approached:[4]

a) Shrinking Encircling Mechanism :
This mechanism is achieved by reducing the value of $\vec{b}$ from 2 to 0 in Eq. (3) as the number of iterations progress. Now, the new position of a search agent can be set or defined anywhere in between or amongst the original position of the agent and the position of the current best agent by assigning arbitrary randomized values for $\vec{B}$ in [-1,1].

b) Spiral Updating Position :
The spiral equation between the position of the whale and the prey which mimics or imitates the helix-shaped movement of the whales is mathematically described as given:

$$\vec{Y}(t+1) = \vec{E'}.e^{cl}.\cos(2\pi l) + \vec{Y^*}(t) \tag{5}$$

From equation (5), the assumed probability is 50% to make a choice between either the shrinking encircling method or the spiral model so as to update the position of whales. The mathematical model is given by equation (6) as follows:

$$\vec{Y}(t+1) = \begin{cases} \vec{Y}(t+1) = \overrightarrow{Y^*(t)} - \vec{B}.\vec{E} & if\ p \leq 0.5 \\ \vec{E'}.e^{cl}.\cos(2\pi l) + \vec{Y^*}(t) & if\ p \geq 0.5 \end{cases} \tag{6}$$

where,
$\vec{E} = \left| \vec{Y^*}(t) - \vec{Y}(t) \right|$ and represents the distance between prey and whale
c is constant for spiral shape,
l is a random number in [-1, 1] and
p describes a randomized number in [0, 1]

3) SEARCH FOR PREY :
The variation of B $^{\rightarrow}$ vector can be used to search for prey, for during the exploration phase. Hence, B $^{\rightarrow}$ can be used with the random values greater than 1 or less than -1 to

force the search agents to move far or away from a reference whale. This is represented mathematically as given by equations (7) and (8) as follows:[4]

$$\vec{E} = \left| \vec{D} . \overrightarrow{Y_{rand}} - \vec{Y} \right| \tag{7}$$

$$\vec{Y}(t+1) = \overrightarrow{Y_{rand}} - \vec{B}.\vec{E} \tag{8}$$

where,
$\overrightarrow{Y_{rand}}$ is defined as random position vector

3 Improved Whale Optimization Algorithm

The improvement of the existing WOA and the development of new Improved WOA known as IWOA is completely focused on the approach followed to generate new population and their updation as iterations progress. This is carried out in the Bubble Net Attacking method where Shrinking Encircling Mechanism and Spiral Updating of position occurs. The idea here is to replace the existing method of Spiral Updation and Search for Prey. This is further carried out in 3 stages namely, Diffusion, Updation–I and Updation-II. The new Updation Process adds the technique of Ranking the Points or Positions such that the Positions or the Points with best rank make it out for the generation of new population of Prey and the Whales. The Gaussian Walk has an upper hand in generating random 2 dimensional fractals compared to Levy Flights and hence Gaussian Walk is preferred.[4-7]

Gaussian Walk :

The Gaussian Walk is also referred to as Random Walk, is a stochastic process. This describes a path or positions in consecutive random sequence or steps based on simple or complex mathematical process.[6]

3.1 Diffusion Process :

The diffusion process used in IWOA implements the strategy of Gaussian Walk as it allows for exchanging information amongst all the involved points or positions. This means that the convergence to the minimum is accelerated alongside producing more favorable and better results. The Gaussian Walk is used for the generation of a Random Walk.

3.2 Updation-I :

This is the first statistical procedure and it ranks all the points individually obtained from the Diffusion process. The ranking system is based on the value of the fitness functions. Here the fitness functions are the 23 Benchmark functions and their description is tabulated in Table 1. The ranking method is based of Uniform Distribution and is applied to every point. Every position or point is first evaluated based on the Fitness function and a probability value is given or simply ranked based on the Uniform Distribution described by equation (9).

$$Ra_{01} = \frac{rank\ (D_i)}{U} \tag{9}$$

where,
$rank\ (D_i)$ is the rank of the Point D_i
U is the number of all the points in the group
Ra_{01} is the Probability value

3.3 Updation – II :

The second Updation process is aimed at modifying the position of point with respect to the change in the positions of other points. This Updation process improves the probability of finding the solution, improves exploration quality and also satisfies the property of diversification. The points from the Fist Updation Process are considered and checked for the conditions and updated further. However, only the new ranks make it to the final process and the latter are discarded.[9-10]

The procedure is two folded and proceeds as follows. Identical to the First Updation Process, if the condition $Ra_{01} < \varepsilon$ is satisfied for a new point say D_i', the current position of D_i' is modified according to the given equations to a new point D_i''.This is given by the two folded equations (10) and (11) as follows :

$$D_i'' = D_i' - \hat{\varepsilon} \times (D_t' - BP) \quad |\varepsilon' \leq 0.5 \tag{10}$$

and

$$D_i'' = D_i' - \hat{\varepsilon} \times (D_t' - D_r') \quad |\varepsilon' > 0.5 \tag{11}$$

where,
D_r' and D_t' are random points selected from the first Updation Process
$\hat{\varepsilon}$ are random numbers by Gaussian Distribution (Gaussian Normal Distribution)
The newly obtained point D_i'' from the second Updation process will once again be replaced back to D_i', if D_i' function's Fitness Value is better than D_i'

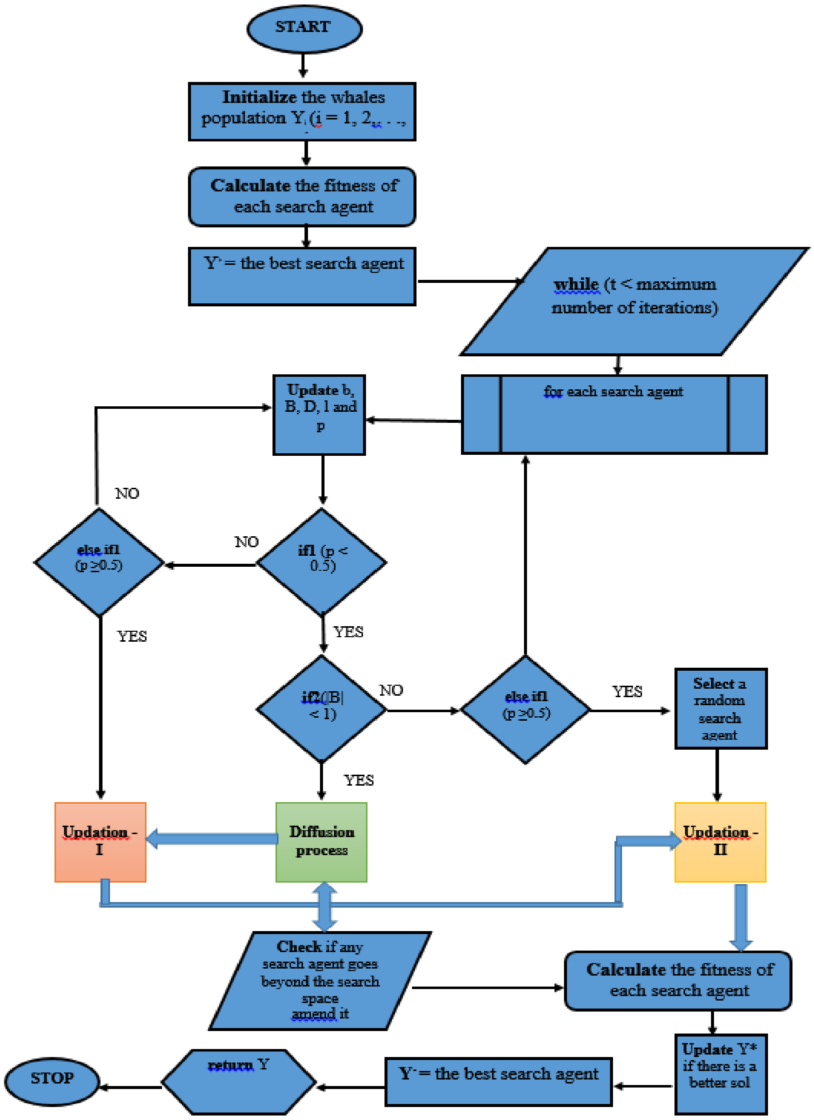

Fig.1: Flow Chart of Improved Whale Optimization Algorithm

4 Results and Discussions

The assessment and evaluation of the obtained results have been based upon the standard benchmark functions. [6-11]

Table 1. List of Benchmark Functions

S.No	FUNCTIONS	DIMENSIONS	RANGE
01.	$F_1(y) = \sum_{i=1}^{n} y_i^2$	10	[-100,100]
02.	$F_2(y) = \sum_{i=1}^{n} \lvert y_i\rvert + \prod_{i-1}^{n} \lvert y_i\rvert$	10	[-10,10]
03.	$F_3(y) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} y_j\right)^2$	10	[-100,100]
04.	$F_4(y) = max_i\{\lvert y_i\rvert, 1 \le i \le n\}$	10	[-100,100]
05.	$F_5(y) = \sum_{i=1}^{n-1} [100(y_{i+1} - y_i^2)^2 + (y_i - 1)^2]$	10	[-30,30]
06.	$F_6(y) = \sum_{i=1}^{n} ([y_i + 0.5])^2$	10	[-100,100]
07.	$F_7(y) = \sum_{i=1}^{n} iy_i^2 + random[0,1])$	10	[-1.28,1.28]
08.	$F_8(y) = \sum_{i=1}^{D} -y_i \sin\left(\sqrt{\lvert y_i\rvert}\right)$	10	[-1.28,1.28]
09.	$F_9(y) = \sum_{i=1}^{n} [y_i^2 - 10\cos(2\pi y_i) + 10]$	10	[-5.12,5.12]
10.	$F_{10}(y) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} y_i^2})$ - $\exp(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi y_i + 1) + 20 + e$	10	[-32,32]
11.	$F_{11}(y) = \frac{1}{4000}\sum_{i=1}^{n} y_i^2 - \prod_{i=1}^{n-1} \cos\left(\frac{y_i}{\sqrt{i}}\right) + 1$	10	[-600,600]
12.	$F_{12}(y) = \frac{\pi}{n}\left\{10\sin(\pi z_1) + \sum_{i=1}^{n-1} (z_i - 1)^2[1 + 10sin^2(\pi z_i + 1)] + (z_n - 1)^2\right\}$	10	[-50,50]
13.	$F_{13}(y) = 0.1\left\{sin^2(3\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2[1 + sin^2(3\pi y_i) + 1]]\right\}$	10	[-50,50]
14.	$F_{14}(y) = \left[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2}(y_i - a_{ij})}\right]^{-1}$	2	[-65.536,65.536]
15.	$F_{15}(y) = \sum_{i=1}^{11} \left[a_i - \frac{y_1(b_i^2 + b_i y_i)}{b_i^2 + b_i y_3 + y_4}\right]^2$	4	[-5,5]
16.	$F_{16}(y) = 4y_1^2 - 2.1y_i^4 + \frac{1}{3}y_1^6 + y_1 y_2$	2	[-5,5]
17.	$F_{17}(y) = \left(y_2 - \frac{5.1}{4\pi^2}y_1^2 + \frac{5}{\pi}y_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos y_1 + 10$	2	[-5,0] , [10,15]
18.	$F_{18}(y) = [1 + (y_1 + y_2 + 1)^2(19 - 14y_1 + 3y_1^2 - 14y_2 + 6y_1 y_2 + 3y_2^2)]$	2	[-2,2]
19.	$F_{19}(y) = -\sum_{i=1}^{4} C_i \exp\left(-\sum_{j=1}^{3} a_{ij}\,(y_j - p_{ij})^2\right)$	3	[0,1]

20.	$F_{20}(y) = -\sum_{i=1}^{4} C_i \exp\left(-\sum_{j=1}^{6} a_{ij}\,(y_j - p_{ij})^2\right)$	6	[0,1]
21.	$F_{21}(y) = -\sum_{i=1}^{5} [(Y - a_i)(Y - a_i)^T + C_i]^{-1}$	4	[0,10]
22.	$F_{22}(y) = -\sum_{i=1}^{7} [(Y - a_i)(Y - a_i)^T + C_i]^{-1}$	4	[0,10]
23.	$F_{23}(y) = -\sum_{i=1}^{10} [(Y - a_i)(Y - a_i)^T + C_i]^{-1}$	4	[0,10]

The comparisons have been carried out on 23 classic Benchmark functions. The Functions from F1 to F7 are Unimodal Benchmark Functions and the functions from F8 to F13 are Multimodal Benchmark Functions, the rest from F14 to F23 are Composite Benchmark Functions. The composite benchmark functions are a combination of Ackley's Function, Sphere Function, Griewank's Function, Rastrigin's Function, Weierstrass Function.

The evaluation is done in Mat Lab and the number of iterations are 1000 and the number of runs are 20. From the obtained results from the Minimum, Maximum, Mean and Standard Deviation values are computed and the values are tabulated. The respective graphs for every benchmark function is plotted. The results are compared to Chaotic Whale Optimization Algorithm (CWOA) [12],and only the best values from different chaotic maps that are included in both papers from the CWOA are considered here.

Table 2. Results

	WOA			IWOA			CWOA	
	MIN	AVG	SD	MIN	AVG	SD	AVG	SD
F1	1.62e-200	1.20e-176	0	1.53e-217	6.57e-200	0	1.68e-70	2.58e-70
F2	5.47e-120	1.72e-113	6.64e-113	1.60e-128	2.66e-121	1.05e-120	3.31e-41	7.46e-41
F3	3.92e-6	1.5393	3.8817	1.74e-42	1.18e-17	5.28e-17	1.14e-292	0.00e+0
F4	1.86e-10	0.1007	0.3016	3.73e-53	1.58e-41	6.83e-41	1.07e-31	1.46e-31
F5	4.882	5.5273	0.335	4.5795	5.3413	0.7244	2.88e+01	4.59e-02
F6	1.23e-06	1.11e-05	9.87e-06	4.92e-07	3.05e-06	2.21e-06	0.00e+0	0.00e+0
F7	4.28e-05	7.77e-04	7.20e-04	5.84e-06	2.25e-04	2.29e-04	n/a	n/a
F8	-4.19e+03	-3.65e+03	638.0025	-4.19e+03	-3.87e+03	421.9218	-1.14e+04	1.62e+03
F9	0	7.11e-16	3.18e-15	0	0	0	0.00e+0	0.00e+0
F10	8.88e-16	3.38e-15	2.03e-15	8.88e-16	2.49e-15	1.81e-15	-1.44e-16	9.86e-21
F11	0	0.0349	0.085	0	0.0418	0.0995	0.00e+0	0.00e+0
F12	1.86e-06	2.48e-05	1.74e-05	1.21e-06	5.71e-06	3.83e-06	n/a	n/a
F13	1.54e-05	1.50e-04	1.83e-04	2.06e-06	5.81e-04	0.0025	n/a	n/a
F14	0.998	1.4445	0.8191	0.998	0.998	2.28e-16	1.91e-02	1.54e-02
F15	3.08e-	6.71e-	3.83e-04	3.08e-04	5.13e-04	3.20e-04	n/a	n/a

	04	04						
F16	-1.0316	-1.031	0	-1.036	-1.031	0	n/a	n/a
F17	0.3979	0.3979	3.66e-07	0.3979	0.3979	3.66e-07	n/a	n/a
F18	3	3	2.24e-06	3	3	2.24e-06	0.00e+00	0.00e+00
F19	-3.8628	-3.861	0.0025	-3.8628	-3.8624	0.0015	8.94e-04	8.36e-04
F20	-3.322	-3.268	0.078	-3.322	-3.248	0.0868	0.00e+00	0.00e+00
F21	-10.153	-9.797	1.6001	-10.1532	-9.8982	1.1399	n/a	n/a
F22	-10.402	-9.236	2.1487	-10.4029	-9.8944	1.598	n/a	n/a
F23	-10.536	-8.086	3.3215	-10.5364	-9.7251	1.9811	n/a	n/a

Here, SD stands for Standard Deviation

From the results, it is obvious that the newly proposed IWOA surpasses WOA in optimizing the functions to their global minima'. Even Compared to CWOA, IWOA outperforms it in most of the cases.

4.1 Comparison Graphs

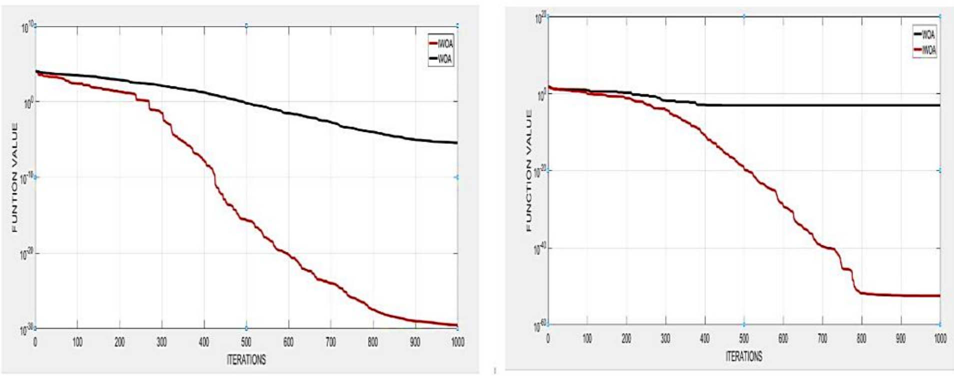

Fig. 2: WOA vs IOA Comparison Graphs for Function F3 and F4

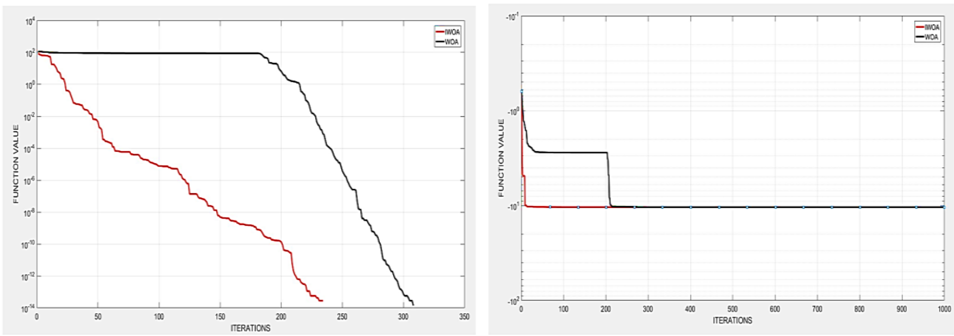

Fig.3: WOA vs IOA Comparison Graphs for Function F9 and F22

From the obtained graphs, it can be seen that for optimization o any solution, here the case being minimization, the Improved Whale Optimization Algorithm (IWOA) outperforms the Whale Optimization Algorithm (WOA).The noteworthy point is the fast rate of Convergence of IWOA compared to WOA. The new ranking system allows for better results as it saves the best obtained values and discards the least scoring values. This also allows for better exploration of the given search space and improves computational efficiency. From Fig. 2, it can be observed the computing efficiency of IWOA over WOA. The function here being to minimization yields the global minima quicker using IWOA. In the Fig. 3 , the speed of IWOA converging to the global minima is quite fast and requires further reduced iterations to achieve global minima compared to the WOA implementation.

5 Conclusion

In this paper, the performance of the existed algorithm that is WOA has been improved by including the updating process. Potential of IWOA is tested on the Benchmark functions with varied complexity, and IWOA outperformed WOA in terms of convergence as well as in the function value. IWOA is performing better than WOA in the exploitation and explorations capabilities. Further IWOA can be used in other optimization problems in different areas.

References

1. X. S. Yang (2010), “ Nature-inspired metaheuristic algorithms “. Luniver Press.
2. X. S. Yang (2010), “ A new metaheuristic bat-inspired algorithm “. *In Nature inspired cooperative strategies for optimization* (NICSO 2010) (pp. 65–74).
3. S. Mirjalili & A. Lewis (2016), “ The whale optimization algorithm. Advances in Engineering Software” , 95, 51–67.
4. M.M. Mafarja & S. Mirjalili(2017), “ Hybrid whale optimization algorithm with simulated annealing for feature selection “. Neurocomputing.
5. P.D.P. Reddy, V.V. Reddy, & T. G. Manohar (2017), “Whale optimization algorithm for optimal sizing of renewable resources for loss reduction in distribution systems “. Renewables: Wind, Water, and Solar, 4(1), 3.
6. A.N. Jadhav, & N. Gomathi (2017), “ WGC: Hybridization of exponential grey wolf optimizer with whale optimization for data clustering “. Alexandria Engineering Journal.
7. A. Kaveh (2017), “ Sizing optimization of skeletal structures using the enhanced whale optimization algorithm “. In Appl. of metaheuristic optimization algorithms in civil engineering (pp. 47–69). Springer.
8. I. Aljarah, H. Faris, & S. Mirjalili (2016), “ Optimizing connection weights in neural networks using the whale optimization algorithm “. Soft Computing, 1–15.
9. T. Liao et al., Ant colony optimization for mixed-variable optimization problems, IEEE. Trans. Evol. Comput. 18 (4) (2014) 503–518.
10. X.-S. Yang, S. Deb, Engineering optimisation by cuckoo search, Int. J. Math. Model. Numer. Optim. 1 (4) (2010) 330–343.
11. M.-Y. Cheng, D. Prayogo, Symbiotic organisms search: a new metaheuristic optimization algorithm, Comput. Struct. 139 (2014) 98–112.
12. G. Kaur, S. Arora, Chaotic whale optimization algorithm, J. of Computational Design and Engineering. 5-2018(275-284)