AN IMPROVEMENT OF HYBRID WHALE OPTIMIZATION ALGORITHM

Mustafa DANACI
ERU Engineering Faculty, Computer Eng. Dept. Kayseri / TURKEY
danaci@erciyes.edu.tr
Bahadur ALIZADA
ERU, Computer Eng., Grad. School of Natural and Applied Sciences bahadur_alizade@yahoo.com

ABSTRACT
The difficulty in solving engineering problems creates difficulties in the selection of the methods to be used. Nature-inspired herd intelligence-based meta-heuristic optimization techniques have recently become the most popular algorithms for solving such problems. In this work, a new hybrid algorithm model has been developed to adapt to various problems. The developed models were adapted to 23 Benchmark test problems in the literature and compared with meta-heuristic algorithms. The algorithms aim to balance the optimization processes of exploration and exploitation. In the development of a meta-heuristic algorithm, it is very difficult to achieve a balance due to its stochastic structure. In this study, the new hybrid model improved by Multi-Verse Optimization (MVO) on the Sine Cosine Whale Optimization Algorithm (SCWOA) hybrid model, which is available in the literature, has increased the success of test problems. Although the SCWOA hybrid balances exploitation and exploration, the MVSCWOA (Multi-Verse Sine Cosine Whale Optimization Algorithm) hybrid algorithm, which was modified by modifying MVO's wormhole existence probability (WEP) and traveling distance rate (TDR), has succeeded in improving this balance further. WEP is used instead of $\boldsymbol{r1}$ parameter, which determines the update direction in SCA, and TDR is used in place of $\boldsymbol{a2}$ (varies between -1 and -2) used in the update of l, which is the inter-element multiplication parameter in WOA. The results obtained from the newly developed hybrid model have shown that it makes the search and exploitation feature more effective by showing better results than SCWOA, WOA, SCA, and MVO. MVSCWOA was successful in test problems.

Keywords: Benchmark, Sine Cosine Algorithm, Whale Optimization Algorithm, Multi-Verse Optimization

1. INTRODUCTION
In recent years, meta-heuristic algorithms have been used more and more to solve, minimize or maximize various problems in almost every aspect of our lives. This is because these algorithms are user-understandable, adaptable to real-life problems, and are simple because they are inspired by events in nature. Examples of famous meta-heuristic algorithms are Genetic Algorithm (GA) [1,2], Ant Colony Optimization (ACO) [3], Particle Swarm Optimization (PSO) [4], Differential Evolution (DE) [5], Evolutionary Programming (EP) [6,7], Artificial Bee Colony (ABC) [8]. According to No Free Lunch (NFL) [9] theorem, it has proved to be not the most appropriate meta-heuristic technique to solve all optimization problems. Therefore, scientists are creating new meta-heuristic optimization techniques by keeping their motivation high: Firefly Algorithm (FA) [10,11], Black Hole (BH) [12], Grey Wolf Optimization (GWO) [13], Cuckoo Search (CS) [14,15], Gravitational Search Algorithm (GSA) [16], Fast Evolutionary Programming (FEP) [17], Whale Optimization Algorithm (WOA) [18], Multi-Verse Optimizer (MVO) [19], Ant Lion Optimizer (ALO) [20], Sine Cosine Algorithm (SCA) [21].

To improve the efficiency of optimization algorithms, various algorithms such as hybrid, optimization, and modification are developed. Through these processes, more successful results are obtained by balancing exploitation and exploration. The missing aspects of two or more algorithms are eliminated by combining advantageous features.

In this research, a previous study in the literature the hybrid Sine Cosine Whale Optimization Algorithm (SCWOA) [22] is developed with the Multi-Verse Optimizer (MVO) algorithm. Through improvement, the balance between exploitation and exploration has been further improved. The improved hybrid algorithm has confirmed itself with test problems. Benchmark functions [23] are used as test problems.

Benchmark functions

Any Benchmark is functional for testing an optimization concept. These aggregate functions can be parsed, differentiated, continuous, discontinuous, scalable, unimodal and multimodal features such as existing algorithms that can be effective information about which kind of problems can be obtained by applying these functions.

The unimodal benchmark does not have a local optimum and has only one global optimum. This makes them very suitable for testing convergence speed and retrieving from algorithms [20]. Detailed information can be made in Table 1 (F1-F7).

Multi-mode benchmarks have multiple local and one global optimizations. The goal of the algorithms is to achieve the best global result without having to stick to the local best result. [20].

If there is a problem in the discovery phase of the algorithm, then a wide search cannot be made. Therefore, the algorithm remains stuck to the local best result, which is not desirable. This is one of the reasons why multimode functions are difficult. [23].

Compared to multimodal functions, fixed-dimension multimodal functions do not allow the number of design variables to be set because of their mathematical formulas but provide a different search field [18]. The properties of the multimodal functions are given in Table 1, respectively n-dimensional (F8-F13) and fixed-dimensional (F14-F23).

At the end of the work, the test plans on three groups of Benchmarks with hybrid characteristics were compared to other meta-heuristic optimization algorithms in the literature.

Table 1. Benchmark functions

Name and Function	Dim	Range	f_{min}
Sphere F1(x)=$\sum_{i=1}^{n} x_i^2$	30	[-100,100]	0
Schwefel2.22 F2(x)=$\sum_{i=1}^{n}\lvert x_i\rvert + \prod_{i=1}^{n}\lvert x_i\rvert$	30	[-10,10]	0
Schwefel1.2 F3(x)=$\sum_{i=1}^{n}(\sum_{j-1}^{i} x_j)^2$	30	[-100,100]	0
Schwefel2.21 F4(x)=$max_i\{\lvert x_i\rvert, 1 \le i \le n\}$	30	[-100,100]	0
Rosenbrock $F5(x) = \sum_{i=1}^{n-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[-30,30]	0
Step F6(x)=$\sum_{i=1}^{n}([x_i + 0.5])^2$	30	[-100,100]	0
Quartic F7(x)=$\sum_{i=1}^{n} ix_i^4 + random[0,1)$	30	[-1.28,1.28]	0
Schwefel F8(x)=$\sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i\rvert})$	30	[-500,500]	-418.9829× Dim
Rastrigin F9(x)=$\sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[-5.12,5.12]	0
Ackley F10(x)=-20exp (-0.2 $\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}$) -exp($\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)$)+20+e	30	[-32,32]	0
Griewank F11(x)=$\frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-600,600]	0
Penalized F12(x)= $\frac{\pi}{n}\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n$ $y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	[-50,50]	0
Penalized2 F13(x) = $0.1\{sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2[1 + sin^2(3\pi x_i + 1)] + (x_n$	30	[-50,50]	0
Foxholes F14(x)= $(\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6})^{-1}$	2	[-65,65]	1
Kowalik F15(x)= $\sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$	4	[-5,5]	0.00030
Camel Six Hump F16(x)=$4x_1^2$-$2.1x_1^4$+$\frac{1}{3}x_1^6$+x_1x_2-$4x_2^2$+$4x_2^4$	2	[-5,5]	-1.0316
Branin RCOS F17(x)= $(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos x_1 + 10$	2	[-5,5]	0.398
GoldStein Price F18(x)= $[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 6x_1x_2 + 3x_2^2)] \times [30 + (2$ $36x_1x_2 + 27x_2^2)]$	2	[-2,2]	3
Hartman3 F19(x)=$-\sum_{i=1}^{4} c_i\exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$	3	[1,3]	-3.86
Hartman6 F20(x)=$-\sum_{i=1}^{4} c_i\exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	6	[0,1]	-3.32
Shekel5 F21(x)=$-\sum_{i=1}^{5}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.1532
Shekel7 F22(x)=$-\sum_{i=1}^{7}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.4028
Shekel10 F23(x)=$-\sum_{i=1}^{10}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.5363

2. RELATED WORKS

2.1. Whale Optimization Algorithm (WOA)

One of the new and successful swarm intelligence-based optimization algorithms introduced by Mirjalili is the Whale Optimization Algorithm (WOA) [18]. The algorithm was created by modeling humpback whale encircling prey; bubble-net attacking, and search for prey behavior. Successful results have been achieved by making improvement by Danacı and Doğan [24], various hybrids by Khalilpourazari and Khalilpourazary [22], Singh and Hachimi [25], Doğan [26] and improvement hybrid model by ALIZADA [27] on WOA.

Encircling prey

Humpback whales encircle it after defining the position of its prey. Since the positions are not predetermined at the beginning, the WOA algorithm considers the best solution for the moment as prey. In other words, hunting is considered to be near optimum. Immediately after the best whale is identified, other whales in the population update their position accordingly. The following equation models this behavior:

$$\vec{D} = \left|\vec{C}.\vec{X}^*(t) - \vec{X}(t)\right| \qquad (1)$$

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A}.\vec{D} \qquad (2)$$

t shows the momentary repetition, the position of the best humpback whale (solution) so far $\vec{X}^*$ provides the position of the best humpback whale obtained so far. $\vec{A}$ and $\vec{C}$ vectors are the specific coefficients.

Bubble-net attacking method (exploitation phase)

The behavior of humpback whales is shown by a mathematical model consisting of two parts. These processes are the shrinking encircling and the spiral update of position respectively.
The bubble hunting technique of the whale is given in the Figure 1.

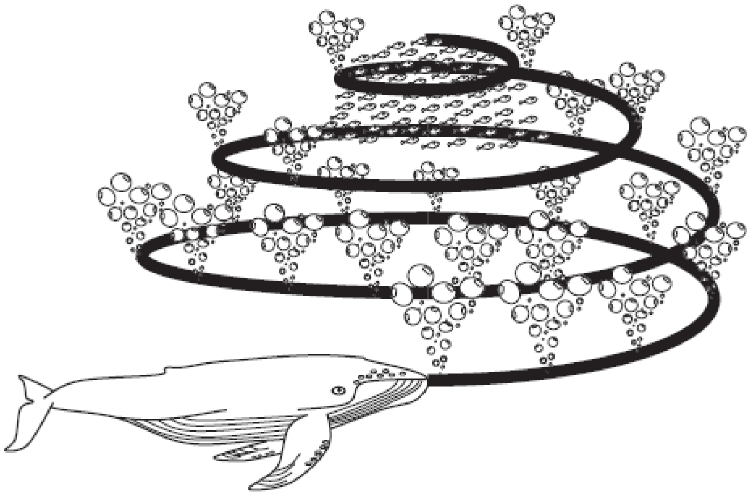

Figure 1. Bubble-net feeding behavior of humpback whales.

Humpback whale floats upward in spiral or helical shape for hunting. It is known that humpback whales in nature perform both narrowing circle or spiral swim simultaneously. Therefore, the algorithm uses both behavior mechanisms with a 50 percent probability. This time can randomly choose the narrowing swim.

$$\vec{X}_{(t+1)} = \begin{cases} \vec{X}^*_{(t)} - \vec{A}.\vec{D} \ if & p < 0.5 \\ \vec{X}^*_{(t+1)} = \vec{D}.e^{bt}.\cos(2\pi l) + X^*_{(t)} \ if \ p \geq 0.5 \end{cases} \qquad (3)$$

Search for prey (exploration phase)

During the exploitation phase, the whale in the population updates its position towards the best whale. But during the exploration phase, an update is made according to a randomly selected whale. The mathematical model of the discovery phase is as follows.

$$\vec{D} = \left|\vec{C}.\vec{X}_{rand} - \vec{X}\right| \tag{4}$$

$$\vec{X}(t+1) = \vec{X}_{rand} - \vec{A}.\vec{D} \tag{5}$$

2.2. Sine Cosine Algorithm (SCA)

One of the meta-heuristic optimization algorithms introduced by Mirjali is the Sine Cosine Algorithm [21]. First, SCO generates a random array of solutions. Next, it chooses the best individual solution to be the target for other solutions according to its objective function value. Each individual in the population updates their position according to the best individual. This update is provided by the following equation.

$$X_i^{t+1} = \begin{cases} X_i^T + r_1 \times \sin(r_2) \times \left|r_3 P_i^t - X_i^T\right|, & r_4 < 0.5 \\ X_i^T + r_1 \times \cos(r_2) \times \left|r_3 P_i^t - X_i^T\right|, & r_4 \geq 0.5 \end{cases} \tag{6}$$

X_i^T is the individual in the t-th iteration population. P_i^t is the position of the best individual ever obtained, and r_1 (update direction), r_2 (update distance), r_3 (stochatic distance identifier), r_4 (chose a sine or cosine)are random numbers.

2.3. Multi-Verse Optimizer (MVO)

One of the physics-based optimization techniques proposed by Mirjalili is the Multi-Verse Optimizer [19]. Algorithm is inspired by the three main concepts: white holes, black holes, and wormholes.

Two important coefficients are considered: wormhole existence probability (WEP) and travel distance ratio (TDR). WEP represents the possibility that the wormhole exists in the universe. The TDR shows the wormhole irradiation distance around the best universe so far. Mathematical formulas for both coefficients are as in (16) and (17), respectively.

$$WEP = min + l \times \left(\frac{max - min}{L}\right) \tag{7}$$

l represents instantaneous iteration; *L* represents maximum iteration.

$$TDR = 1 - \frac{l^{1/p}}{L^{1/p}} \tag{8}$$

p defines the accuracy of exploitation. High value means that it has a good phase of exploitation.

2.4. Improved Hybrid Model

SCWOA, which has already been introduced to the literature, is a hybrid model that balances both by taking into account the good search capability of the WOA and the good exploitation feature of the SCA. In this study, a triple hybrid has been developed by making a new improvement on the existing hybrid model by using the Multi-Verse Optimizer features. The motivation source, mathematical model and pseudocode of the model are given below.

SCWOA, a hybrid of SCA and WOA previously found in the literature, is an algorithm that has managed to balance both search and exploitation characteristics. In the study, some of the features of MVO were replaced with the features found in SCWOA. Two features were used in the study. These are the WEP and TDR found in the multi-verse algorithm. WEP is used instead of the parameter r_1, which determines the update direction in SCA, and TDR is replaced by a_2 (ranging from -1 to -2), which is used for the update of the inter-element multiplication parameter *l* in WOA. The results from the newly developed hybrid model have proven to be more effective than SCWOA, WOA, SCA, and MVO, making search and exploitation characteristics more efficient.

4.1. Proposed Mathematical model

In the created model, the function that occurs when the SCA updates operator is replaced by the narrowing containment mechanism of WOA is indicated below.

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 \times \sin(r_2) \times \left| r_3 X_{(t)}^* - X_i^t \right|, & q < 0.5 \\ X_i^t + r_1 \times \cos(r_2) \times \left| r_3 X_{(t)}^* - X_i^t \right|, & q \geq 0.5 \end{cases} \quad (9)$$

Replicas r_1 and $a_2$are also replaced by WEP and TDR that run on iterations. These are more clearly understood in the pseudocode.

Pseudocode

```
Initialize random population.
Calculate the fitness of each search agent.
The best search agent = X* .
While the last criterion is not met
for each search agent
Update A, l, p, WEP Eq. (7), TDR Eq. (8), r2, r3, and r4
if1(p<0.5)
        if2(r4<0.5)
Update the position of the current search agent by the Eq. (9)
        else if2(r4≥ 0.5)
                Select a random search agent
Update the position of the current search agent by the Eq. (9)
        end if2
else if1(p≥0.5)
Update the position of the current search by the Eq. (3)
end if1
end for
Check if any search agent goes beyond the search space and amend it
Calculate the fitness of each search agent
Update X* if there is a better solution
t=t+1
end while
Return X*
```

5. RESULTS

In the MVSCWOA experiment, the number of populations was 30; the number of iterations was 500, and the dimensions of the Benchmark function were adjusted by setting as previously stated in the Benchmark function tables. Each function was run 30 times, and the mean of the obtained values were calculated. Comparison time MVSCWOA, SCWOA, SCA, MVO algorithms were run. The MVO Matlab code is available from Mirjalili's open-source code. Benchmarking time was run with the same parameters. From the article about WOA, the results of WOA, PSO, GSA, and FEP algorithms were compared and a total of seven algorithms were compared. The results are shown in Table 2.

When the results of the MVSCWOA algorithm are examined, the results of F9 over performance, F1-F4, F7, F9-F11, F16, and F21 functions are very good. Of these, F1-F4, F9-F10, and F16 showed better results than all algorithms in comparison, F7, F11, F21 showed second place in comparison. In other functions showed average and approximate results. Overall, an average improvement was made.

Table 2. Performance analysis with benchmark function results

M	F	f_{min}	MVSCWOA	1-SCWOA	2-WOA	3-SCA	4-MVO	5-PSO	6-GSA	7-FEP
			ave							
UNIMODAL	F1	0	**3.6035E-65**	2.5662E-30	1.41E −30	17.454462	1.251655	0.000136	2.53E-16	0.00057
	F2	0	**1.9681E-34**	1.2455E-16	1.06E −21	0.018628	9.913015	0.042144	0.055655	0.0081
	F3	0	**1.6432E-55**	4.8624E-27	5.39E −07	10016.544	245.75308	70.12562	896.5347	0.016
	F4	0	**1.6259E-32**	7.3999E-16	0.072581	32.825737	1.782515	1.086481	7.35487	0.3
	F5	0	27.91003	27.2007	27.86558	48385.494	298.2843	96.71832	67.54309	**5.06**
	F6	0	2.569737	0.232776	3.116266	19.44915	1.256098	0.000102	2.5E-16	**0**
	F7	0	**0.000333**	**0.000316**	0.001425	0.09472	0.037924	0.122854	0.089441	0.1415
MULTIMODAL	F8	-12569.487	-5167.9574	-11515.5	-5080.76	-3712.787	-7844.6521	-4841.29	-2821.07	**-12554.5**
	F9	0	**0**	**0**	**0**	349.87198	123.0991	46.70423	25.96841	0.046
	F10	0	**8.8818E-16**	8.3489E-15	7.4043	15.77695	1.951047	0.276015	0.062087	0.018
	F11	0	**4.996E-16**	**0**	0.000289	1.049631	0.853236	0.009215	27.70154	0.016
	F12	0	0.191217	0.010909	0.339676	548.78755	1.867438	0.006917	1.799617	**9.2E-06**
	F13	0	1.934423	0.1918	1.889015	69343.16	0.213151	0.006675	8.899084	**0.00016**
FIXED-DIMENSION MULTIMODAL	F14	1	5.718113	1.626363	2.111973	1.925526	**0.998**	3.627168	5.859838	1.22
	F15	0.00030	0.000593	0.000674	0.000572	0.001002	0.004801	0.000577	0.003673	**0.0005**
	F16	-1.0316	**-1.0316**	**-1.0316**	-1.03163	-1.03156	**-1.0316**	-1.03163	-1.03163	-1.03
	F17	0.398	0.397949	0.397968	0.397914	0.399486	0.39789	0.397887	0.397887	**0.398**
	F18	3	3.000453	3.00057	**3**	3.000093	11.1	**3**	**3**	3.02
	F19	-3.86	-0.30048	-0.30048	-3.85616	-0.30048	-0.30048	-3.86278	-3.86278	**-3.86**
	F20	-3.32	-3.23337	-3.21142	-2.98105	-2.69072	-3.27712	-3.26634	**-3.31778**	-3.27
	F21	-10.1532	**-8.22838**	**-8.41452**	-7.04918	-2.874981	-6.70704	-6.8651	-5.95512	-5.52
	F22	-10.4028	-8.34166	-8.8921	-8.18178	-3.160536	-8.570013	-8.45653	**-9.68447**	-5.53
	F23	-10.5363	-8.677407	-9.51277	-9.34238	-3.552445	-9.12424	-9.95291	**-10.5364**	-6.57

Best result   Secondary Best result

The selected parameter settings make the problems more challenging, allowing hybrid algorithms to perform more easily. This can be generally evaluated by selecting the size as 30 in n-dimensional Benchmark functions. Dimensions, search area, population size, iteration number were determined according to various studies and selected articles. Since the algorithms operate with random logic, averaging 30 times, the code is run for each function. Thus, the result becomes more pronounced.

Although the SCWOA hybrid balances exploitation and exploration, MVO's MVSCWOA hybrid algorithm, which was developed using MVO's WEP and TDR, has further improved this balance.

The performance of the newly developed MVSCWOA hybrid algorithm has been compared. MVSCWOA was successful in 9 out of 23 test problems. According to the test results, it showed good results in both unimodal and multimodal test problems. This shows that the developed model has improved the convergence rate further and finds the global optimum without being attached to the local optimum.

6. CONCLUSION

In this study, the performance analysis was evaluated by adapting the developed hybrid algorithm to 23 Benchmark functions for testing purposes. The test functions used differ from each other in terms of features, and comprehensive evaluation is obtained. Compared to the algorithms used in the study and the results of some popular optimization algorithms used in the comparison, the developed hybrid model was observed to be quite successful.

Each of the SCA, WOA, and MVO algorithms affect the results according to their optimization process. Although SCA is good at exploitation, it is not good at exploration. WOA is the opposite. MVO has both exploitation and discovery features. The hybrid model developed in the study aimed to achieve better results by combining the features, and it was found to produce highly successful and competitive results. The performance of the hybrid algorithms can be examined in a wider field by adapting the experiments to future problems such as traveling salesman problem (GSP), real-life problems and engineering problems. Hybridizations can also be performed in different tests in terms of exploitation and exploration.

REFERENCES

[1] Holland JH. Genetic algorithms. Sci Am 1992;267:66–72.
[2] Holland JH, Reitman JS. Cognitive systems based on adaptive algorithms. ACM SIGART Bull 1977. p. 49–49.
[3] Colorni A, Dorigo M, Maniezzo V. Distributed optimization by ant colonies. In: Proceedings of the first European conference on artificial life; 1991. p. 134–42.
[4] Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization (PSO). In Proc. IEEE Int. Conference on Neural Networks, Perth, Australia (pp. 1942-1948).
[5] Storn R, Price K. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 1997;11:341–59.
[6] Fogel LJ, Owens AJ, Walsh MJ. Artificial intelligence through simulated evolution; 1966.
[7] Yao X, Liu Y, Lin G. Evolutionary programming made faster. IEEE Trans Evol Comput 1999; 3:82–102.
[8] Basturk B, Karaboga D . An artificial bee colony (ABC) algorithm for numeric function optimization. In: Proceedings of the IEEE swarm intelligence symposium; 2006. p.12–14
[9] Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE Trans Evol Comput 1997;1:67–82.
[10] Yang X-S. Firefly algorithm, Levy flights and global optimization. In: Research and development in intelligent systems XXVI. Springer; 2010. p. 209–18.
[11] Yang X-S. Firefly algorithm, stochastic test functions and design optimisation. Int J Bio-Inspired Comput 2010;2:78–84.
[12] Hatamlou A . Black hole: a new heuristic optimization approach for data clus- tering. Inf Sci 2013; 222:175–84 .
[13] Mirjalili Seyedali, Mirjalili Seyed Mohammad, Lewis Andrew. Grey wolf optimizer. Adv Eng Software 2014;69:46–61.
[14] Yang X-S, Deb S. Cuckoo search via Lévy flights. In: World congress on nature & biologically inspired computing, 2009. NaBIC 2009; 2009. p. 210–4.
[15] Yang X-S, Deb S. Engineering optimisation by cuckoo search. Int J Math Model Numer Optim 2010;1:330–43.
[16] Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: a gravitational search algorithm. Information sciences, 179(13), 2232-2248.
[17] Yao, X., & Liu, Y. (1996). Fast Evolutionary Programming. Evolutionary programming, 3, 451-460.
[18] Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. Advances in engineering software, 95, 51-67.
[19] Mirjalili, S., Mirjalili, S. M., & Hatamlou, A. (2016). Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Computing and Applications, 27(2), 495-513.
[20] Mirjalili, S. (2015). The ant lion optimizer. Advances in Engineering Software, 83, 80-98.
[21] Mirjalili, S. (2016). SCA: a sine cosine algorithm for solving optimization problems. Knowledge-Based Systems, 96, 120-133.
[22] Khalilpourazari, S., & Khalilpourazary, S. (2018). SCWOA: an efficient hybrid algorithm for parameter optimization of multi-pass milling process. Journal of Industrial and Production Engineering, 35(3), 135-147.
[23] Jamil, M., & Yang, X. S. (2013). A literature survey of Benchmark functions for global optimization problems. arXiv preprint arXiv:1308.4008.
[24] Danacı, M,.Doğan, C. (2019, Nisan). Improved Whale Optimization Algorithm. Talas, M *Uluslararası Erciyes Bilimsel Araştırmalar Kongresi* (s482-496)Yer: Erciyes üniversitesi
[25] Singh, N., & Hachimi, H. (2018). A new hybrid whale optimizer algorithm with mean strategy of grey wolf optimizer for global optimization. Mathematical and Computational Applications, 23(1), 14.
[26] Doğan, C (2019). Balina Optimizasyon Algoritması ve Gri Kurt Optimizasyonu algoritmaları kullanılarak yeni hibrit optimizasyon algoritmalarının geliştirilmesi. Erciyes Üniversitesi/Fen Bilimleri Enstitüsü, Kayseri

[27] ALIZADA, B. (2019) Sürü tabanlı karınca aslanı ve balina optimizasyonu algoritmalarının fizik tabanlı algoritmalarla Hibritleştirilmesi.(Yüksek lisans tezi) Erciyes Üniversitesi / Fen Bilimleri Enstitüsü, Kayseri.