INTERNATIONAL SCIENTIFIC AND VOCATIONAL JOURNAL (ISVOS JOURNAL)

Vol.: 4 Issue: 1 Date: June 2020 Received: 18.12.2019 Accepted: 10.05.2020 Final Version: 29.06.2020

ISVOS Journal, 2020, 4(1): 21-30

Improved Whale Optimization Algorithm Based On π Number

Bahadur Alizada [a,1]

[a] Erciyes University, Engineering Faculty, Computer Engineering, 38000, Kayseri, Turkey
ORCID ID: 0000-0001-6587-4057

Abstract

In this study, an improved version is presented as a result of experiments performed on the whale optimization algorithm (WOA) in the literature. As a result of the experiments, π number was added to the $\vec{C}$ coefficient vector of the algorithm. The developed WOA algorithm based on the number of π was adapted to test problems. The 23 most common Benchmark functions have been selected as test problems. In line with the results, it was observed that the exploitation and exploration phases of the WOA developed. The success of the results has proven itself in comparison with other algorithms.

Keywords: "Whale Optimization Algorithm, Benchmark Functions,Optimization"

1. Introduction

The use of meta-heuristic optimization techniques is becoming more common day by day. The most important reasons for this are nature-inspired simple structured algorithms, a comfortable control mechanism by scientists and users, and can be easily adapted to various real-life problems. Meta-heuristic methods are preferred in terms of low cost of calculation and time saving compared to classical methods.

Meta-heuristic optimization algorithms can be varied according to the inspiration on which it is based. Table 1 shows this variation. The table includes both popular and new and successful meta-heuristic optimization algorithms. Of course, the following list covers very few of the meta-heuristic techniques.

Table 1. Classification of the meta-heuristic methods

Meta-heuristic optimization techniques				
Evolutionary inspired	Physics inspired	Swarm inspired	Human inspired	Other
Genetic Algorithm (GA) [1]	Simulated Annealing (SA) [4-5]	Ant Colony Optimization (ACO) [8]	Firework Algorithm (FA) [11]	Sine Cosine Algorithm (SCA) [16]
Genetic Programming (GP) [2]	Gravitational Search Algorithm (GSA) [6]	Particle Swarm Optimization (PSO) [9]	Taboo Search (TS) [12-14]	Stochastic Fractal Search (SFS) [17]
Differential Evolution (DE) [3]	Multi-Verse Optimizer (MVO) [7]	Artificial Bee Colony (ABC) [10]	Harmony Search (HS) [15]	Water Cycle Algorithm (WCA) [18]

According to No Free Lunch (NFL) [19] the reason for the increase in the variety of optimization algorithms [20-21] is that there is no adaptive method that can be resolve all kinds of problems. The type of problem to be adapted varies depending on the structure of the algorithms. Based on this motivation, scientists develop new, hybrid, improvement, modified optimization techniques.

The meta-heuristic optimization algorithm based on WOA [22] herd intelligence was used. Although the algorithm is new, it achieves very successful results. There are several improved models in the literature. Table 2 gives some examples of improved models.

[1] Sorumlu Yazar. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000 .
E-posta adresi: …………

Table 2. Improved whale optimization algorithms lists

Improved Model	Applied Problem	Year of proposal
An adaptive WOA [23]	Global optimization	2016
WOA with inertia weight [24]	Global optimization	2016
Enhanced WOA [25]	Sizing Optimization	2016
Improved WOAs based on inertia weights [26]	Global optimization	2017
Lévy Flight Trajectory-Based WOA [27]	Global optimization	2017
An improved chaotic WOA [28]	Parameter estimation of photovoltaic cells	2017
Multi-Objective WOA [29]	Wind speed forecasting	2017
An improved Lévy based WOA [30]	Bandwidth-efficient virtual machine placement	2018
Chaotic WOA [31]	Global optimization	2018
An improved WOA [32]	Global optimization	2018
Non-dominated sorting Multi-Objective WOA [33]	Content-based image retrieval	2018
An improved WOA [34]	PV models	2018
A modified WOA [35]	0–1 knapsack problem	2019
An improved hybrid WOA [36]	Global optimization	2019

In this study, the WOA $\vec{C}$ coefficient vector was updated based on the number π, inspired by the number of SCA update distances. After the update, the algorithm was tested and adapted to the 23 most used Benchmark problems (functions) in the literature. The parametric properties of the functions are showed in Table 3. The obtained results are proved by comparing the WOA and other population algorithms.

Table 3. Features of benchmark problems

Function	Dim	Range	f_{min}
$F_1(x) = \sum_{i=1}^{n} x_i^2$	30	[−100,100]	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	[−10, 10]	0
$F_3(x) = \sum_{i=1}^{n} (\sum_{j-1}^{i} x_j)^2$	30	[−100,100]	0
$F_4(x) = \max_i \{\lvert x_i \rvert, 1 \le i \le n\}$	30	[−100,100]	0
$F_5(x) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$	30	[−30, 30]	0
$F_6(x) = \sum_{i=1}^{n} (\lvert x_i + 0.5 \rvert)^2$	30	[−100,100]	0
$F_7(x) = \sum_{i=1}^{n} i x_i^4 + random[0,1)$	30	[−1.28, 1.28]	0
$F_8(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$	30	[−500, 500]	−12569.487
$F_9(x) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$	30	[−5.12, 5.12]	0
$F_{10}(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e$	30	[−32, 32]	0
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$	30	[−600, 600]	0
$F_{12}(x) = \frac{\pi}{n}\left\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2\left[1 + 10\sin^2(\pi y_{i+1})\right] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$	30	[−50, 50]	0
$y_i = 1 + \frac{x_i + 1}{4} \quad u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$			
$F_{13}(x) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2\left[1 + \sin^2(3\pi x_i + 1)\right] + (x_n - 1)^2\left[1 + \sin^2(2\pi x_n)\right]\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	[−50, 50]	0
$F_{14}(x) = (\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6})^{-1}$	2	[−65, 65]	1
$F_{15}(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$	4	[−5, 5]	0.0003
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	[−5, 5]	−1.0316
$F_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos x_1 + 10$	2	[−5, 5]	0.398
$F_{18}(x) = \left[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right] \times$ $\left[30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\right]$	2	[−2, 2]	3
$F_{19}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$	3	[1, 3]	−3.86
$F_{20}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	6	[0, 1]	−3.32
$F_{21}(x) = -\sum_{i=1}^{5}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	−10.1532
$F_{22}(x) = -\sum_{i=1}^{7}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	−10.4028
$F_{23}(x) = -\sum_{i=1}^{10}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	−10.5363

2. Related Works

2.1. Whale optimization algorithm

WOA is a swarm-based optimization technique. The method was inspired by the acts of humpback whales.

The location of the optimum design in the area where the encircling pray is first searched by the whale is unknown. Therefore, the algorithm assumes that the marked pray is the best solution at the moment. This means that the solution is near optimum value. Then the best searching agent among the whales is selected. Depending on this selection, other population whales update their position relative to the best whale. This update technique is defined by the following equation:

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

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

$t$is instantaneous iteration. $\vec{X}^*$ is the location of the best individual ever achieved. $\vec{X}$ is the location vector. $\vec{A}$and $\vec{\boldsymbol{C}}$ vectors are the specific coefficients. These coefficients are determined by the following equations.

$$\vec{A} = 2\vec{a}.\vec{r} - \vec{a} \tag{3}$$

$$\vec{C} = 2.\vec{r} \tag{4}$$

$\vec{a}$ is a parameter whose initial value decreases linearly from 0 to 2 during iterations. $\vec{r}$ gets a random values between 0 and 1.

The bubbly attack represents the phase of exploitation in meta-heuristic optimization techniques. Humpback whales perform the attack with two mechanisms of both shrinking containment and curled updating of position. Since the humpback whale in nature performs both, the modeling algorithm is given as follows:

$$\vec{X}_{(t+1)} = \begin{cases} \vec{X}^*_{(t)} - \vec{A}.D & if \quad p < 0.5 \\ \vec{X}^*_{(t+1)} = \vec{D}.e^{bt}.\cos(2\pi l) + \vec{X}^*_{(t)} & if \quad p \geq 0.5 \end{cases} \tag{5}$$

b defines shape of the fixed value logarithmic curled. $\vec{l}$ gets a random numbers between -1 and 1.

The encircling technique represents the exploration phase in meta-heuristic optimization techniques. During the exploitation phase, the update was best made according to the position of the whale. In the discovery phase, this update is performed randomly. These time model equations are expressed as follows:

$$\vec{D} = \left|\vec{C}.\vec{X}_{rand} - \vec{X}\right| \tag{6}$$

$$\vec{X}(t+1) = \vec{X}_{rand} - \vec{A}.\vec{D} \tag{7}$$

$\vec{X}_{rand}$ represents the randomly selected position vector (the position of the whale) from the population.

2.2. Sine cosine algorithm

The meta-heuristic optimization technique, SCA, is inspired by the graphical movements of the sine and cosine. SCA first generates random solutions. Then, it chooses the best individual solution based on its suitability value. Then the individuals in the population update their current position according to the best resultant with the help of the following equation:

$$X_i^{t+1} = \begin{cases} X_i^T + r_1 \times sin(r_2) \times |r_3 P_i^t - X_i^T|, r_4 < 0.5 \\ X_i^T + r_1 \times cos(r_2) \times |r_3 P_i^t - X_i^T|, r_4 \geq 0.5 \end{cases} \tag{8}$$

X_i^T is the instantaneous location. P_i^t is the location of the best individual, r_1, r_2, r_3, r_4 are random variables, respectively the direction of update, the update distance, the weight of the target, the balance between sine and cosine movements.

2.3. The proposed improvement

In this study, it is inspired by the random number which represents the update distance r_2 in the $[0,2\pi]$ range in the SCA. The $\vec{C}$ parameter in the original WOA ranges from$[0,2]$. By adding only the number π as a multiplier, you can increase the range of changes to achieve more successful results. Therefore, the model (4) is updated by taking the following figure.

$$\vec{C} = 2.\pi.\vec{r} \quad (9)$$

Since the $\vec{C}$ specific number is used in both encircling pray (exploitation phase) and hunting search (reconnaissance phase) equations, the effect is two-fold. In this way, the balance between exploitation and exploration is strengthened and better results are obtained.

3. Results and Discussion

In this study, the comparison with the original algorithm was performed to show the improvement in numerical efficiency. 23 classical Benchmark functions [37-40] were chosen as test problems. For comparison, the results used in the first article of WOA [32] were used.

Only the developed algorithm (PIWOA) and original algorithm (WOA) codes were run and the results were obtained. Other results are taken from the article. In all algorithms, population size was 30 and iteration was 500. The average of the first 30 results obtained in each function when executing the codes is given in Table 6. Table 5 shows the minimum and maximum values. The function parameters are set according to the previously given Table 2.

First, a few tests were made to increase or decrease the number of pi. In Table.4, experiments were done on some coefficients of π and the results were compared. If the C coefficient value increases or decreases, an experiment has been conducted on how the behavior of the algorithm will change.

Table 4. Comparison of coefficients of πaccording to benchmark functions

	PIWOA					
F	π	2 π	3 π	π /2	1/ π	1/2 π
	ave					
F_1	4,12E-123	5.55E-133	6,79E-130	2,56E-98	8,97E-87	3,04E-92
F_2	2,16E-73	1,53E-80	2,05E-78	5,39E-61	1,34E-58	4,93E-63
F_3	66758,07	78056,76	79318,79	52968,93	66132,46	71551,06
F_4	52,84312	52,03767	60,75798	41,815	56,23557	54,64768
F_5	28,34118	28,59631	28,66984	27,96659	28,17009	28,14091
F_6	0,167757	0,154046	0,188019	0,206281	0,578550	0,676079
F_7	0,001883	0,002604	0,001001	0,002192	0,002285	0,001905
F_8	-12355,79	-12326,78	-12377	-12003,62	-10318,97	-10881,84
F_9	0	3,79E-15	0	5,68E-15	0	1,90E-15
F_{10}	2,78E-15	2,78E-15	3,14E-15	3,26E-15	4,44E-15	3,26E-15
F_{11}	0	0	0	0,009211	0	0
F_{12}	0,009849	0,014054	0,011485	0,014154	0,034508	0,029392
F_{13}	0,223656	0,189085	0,145408	0,271735	0,803409	0,767341
F_{14}	2,6343	3,484245	2,873919	2,407947	3,579533	4,072622
F_{15}	0,000887	0,000849	0,000863	0,001065	0,000699	0,001265
F_{16}	-1,0316	-1,0316	-1,0316	-1,0316	-1,0316	-1,0316
F_{17}	0,397911	0,397926	0,397923	0,397897	0,397915	0,397946
F_{18}	3	6,611693	6,780357	3	3,00122	3
F_{19}	-3,83724	-3,83192	-3,81011	-3,85344	-3,85262	-3,85745
F_{20}	-3,16966	-3,12847	-3,10678	-3,22981	-3,17878	-3,18244
F_{21}	-9,98939	-9,70178	-9,77219	-9,40376	-8,51101	-8,95557
F_{22}	-10,2779	-10,1736	-10,0866	-9,19885	-7,70737	-8,79525
F_{23}	-10,3163	-10,2917	-10,2185	-9,32655	-7,44693	-8,90413

The results in Table 4 show that only the use of $\boldsymbol{\pi}$ gives more successful results. Reducing or increasing the coefficient factors has only made progress in certain functions and in some cases has not even achieved the average result. Looking at the table, $\boldsymbol{\pi}$

proves that it is the test that shows the best results in either double or general comparison. Therefore, in the next tables, the results are compared with the π coefficient.

Table 5. Comparison of PIWOA and WOA algorithms according to benchmark functions

F	PIWOA			WOA		
	ave	*max*	*min*	*ave*	*max*	*min*
F_1	4,12E-123	9.0613E-139	1.2133E-121	1,61E-73	2,579E-87	3,037E-72
F_2	2,16E-73	1.2036E-81	6.4822E-72	3,71E-50	4,7167E-57	1,0097E-48
F_3	66758,07	41603.4946	90989.3035	43621,36	16992,4815	68700,9912
F_4	52,84312	2.2999	91.183	46,01006	0,24429	84,8424
F_5	28,34118	27.6717	28.7321	27,96338	26,9892	28,7592
F_6	0,167757	0.043911	0.2995	0,39663	0,0973	0,8498
F_7	0,001883	8.421E-06	0.011	0,003941	0,00011573	0,014392
F_8	-12355,79	-12569.47	-11558.36	-10138,55	-12567,2305	-7438,5198
F_9	0	0	0	7,58E-15	0	1.1369E-13
F_{10}	2,78E-15	8.8818E-16	7.9936E-15	4,2E-15	8.8818E-16	7.9936E-15
F_{11}	0	0	0	0	0	0
F_{12}	0,009849	0.0026526	0.031354	0,023297	0,0049458	0,11166
F_{13}	0,223656	0.078179	0.5647	0,448593	0,07261	1,1382
F_{14}	2,6343	0.998	10.7632	2,959695	0.998	10,7632
F_{15}	0,000887	0.00031518	0.0022519	0,000602	0,00030782	0,0016208
F_{16}	-1,0316	-1.0316	-1.0316	-1,0316	-1.0316	-1.0316
F_{17}	0,397911	0.39789	0.39795	0,397893	0,39789	0,3979
F_{18}	3	3	3.0004	3	3	3.0003
F_{19}	-3,83724	-3.8626	-3.7187	-3,85732	-3,8628	-3,8344
F_{20}	-3,16966	-3.3208	-1.8276	-3,1939	-3,3219	-1,8403
F_{21}	-9,98939	-10.152	-9.2686	-8,01905	-10,1523	-2,6256
F_{22}	-10,2779	-10.4012	-9.7117	-7,50882	-10,4008	-1,8355
F_{23}	-10,3163	-10.5337	-9.2164	-6,9726	-10,5358	-2,4216

The results in the Table 5 showed that the PIWOA was more successful than the WOA. Of the 23 test problems, 16 showed the best results, while the others showed approximate results. This proves that the improvement is going well.

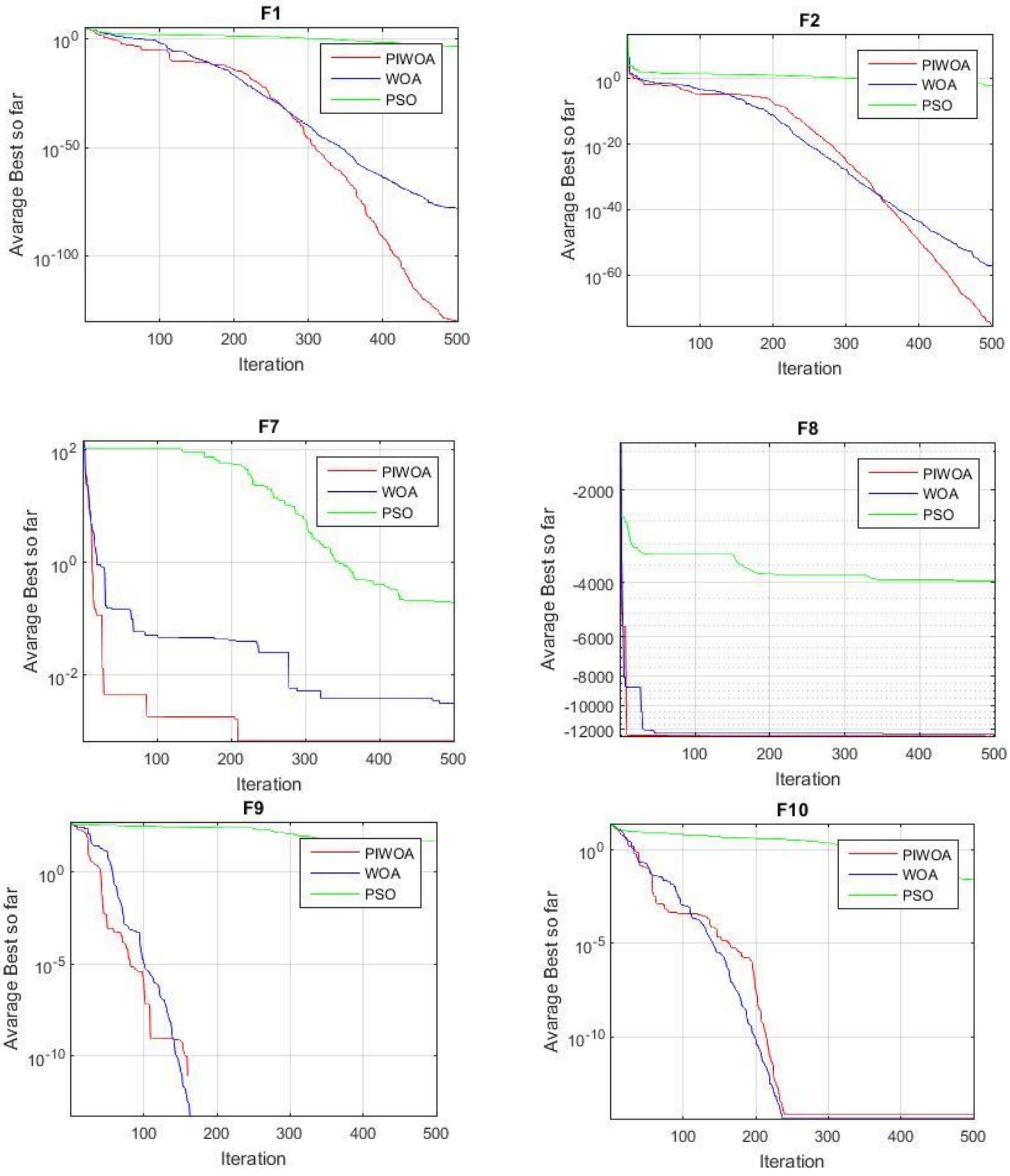
Table 6. Comparison of algorithms according to benchmark functions

F	PIWOA	WOA	PSO	GSA	FEP	DE
			ave			
F_1	4,12E-123	1,61E-73	0.000136	2.53E-16	0.00057	8.2E −14
F_2	2,16E-73	3,71E-50	0.042144	0.055655	0.0081	1 .5E −09
F_3	66758,07	43621,36	70.12562	896.5347	0.016	6 .8E −11
F_4	52,84312	46,01006	1.086481	7.35487	0.3	0
F_5	28,34118	27,96338	96.71832	67.54309	5.06	0
F_6	0,167757	0,39663	0.000102	2.5E-16	0	0
F_7	0,001883	0,003941	0.122854	0.089441	0.1415	0 .00463
F_8	-12355,79	-10138,55	-4841.29	-2821.07	-12554.5	−11080.1
F_9	0	7,58E-15	46 .70423	25 .96841	0.046	69.2
F_{10}	2,78E-15	4,2E-15	0 .276015	0 .062087	0.018	9 .7E −08
F_{11}	0	0	0 .009215	27 .70154	0.016	0
F_{12}	0,009849	0,023297	0 .006917	1 .799617	9.2E-06	7 .9E −15
F_{13}	0,223656	0,448593	0 .006675	8 .899084	0.00016	5 .1E −14
F_{14}	2,6343	2,959695	3.627168	5.859838	1.22	0 .998004
F_{15}	0,000887	0,000602	0.000577	0.003673	0.0005	4 .5E −14
F_{16}	-1,0316	-1,0316	-1.03163	-1.03163	-1.03	−1 .03163
F_{17}	0,397911	0,397893	0.397887	0.397887	0.398	0 .397887
F_{18}	3	3	3	3	3.02	3
F_{19}	-3,83724	-3,85732	-3.86278	-3.86278	-3.86	N/A
F_{20}	-3,16966	-3,1939	-3.26634	-3.31778	-3.27	N/A
F_{21}	-9,98939	-8,01905	-6.8651	-5.95512	-5.52	−10 .1532
F_{22}	-10,2779	-7,50882	-8.45653	-9.68447	-5.53	−10 .4029
F_{23}	-10,3163	-6,9726	-9.95291	-10.5364	-6.57	−10 .5364

Table 6 shows the comparison of algorithms. Comparison time WOA, PSO, GSA, Fast Evolutionary Programming (FEP), DE results were used. It should be emphasized that the results of the PSO, GSA, FEP, DE algorithms are taken from the WOA article [32].

Unimodal benchmarks (F1-F7) should be reviewed to test the exploitation phase of the algorithm. The time of comparisons was successful in all of the functions F1, F2 and F7. F5 and F6 are average values. Its success on unimodals show that the PIWOA has high exploitation ability.

The discovery phase of the algorithm is tested by multimodal benchmarks (n-dimensional and fixed-dimension) (F8-F23). According to the results, 6 of them showed better results than all of them. Others achieved success with approximate and average results. It has achieved the success of the original algorithm. Apart from this, it has assumed DE in some places and showed approximate values in some places. Thus, it proved to be a competitive model.



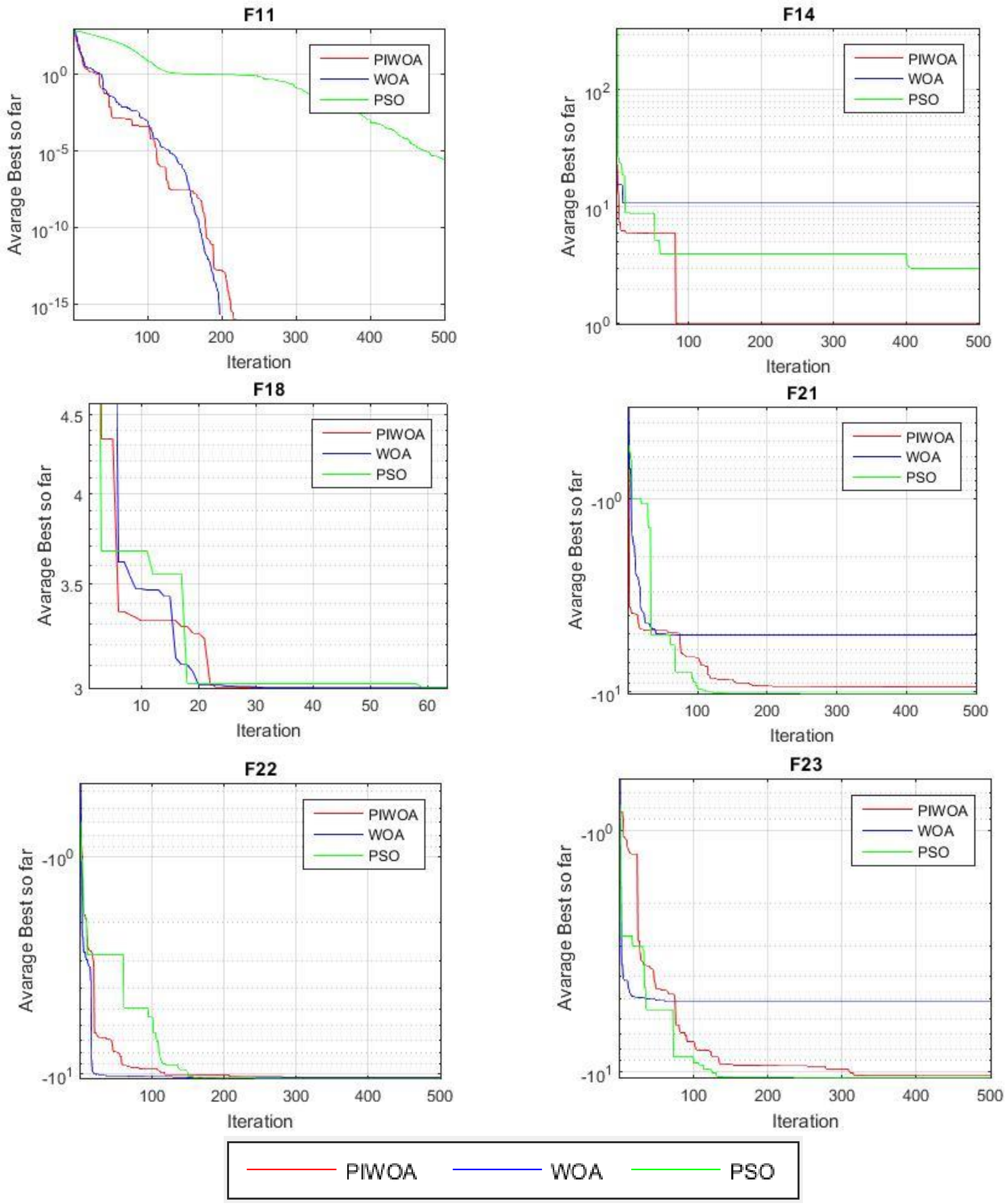


Figure 1. Comparison of convergence curves of PIWOA, WOA, and PSO obtained in some of the functions.

WOA and PSO algorithms were used to compare the convergence curve. In F1 and F2, convergence was gradual throughout the iterations. In F7, F9, F10, F11, F21, F22, F23, although this progressed to a certain place, it suddenly converged rapidly. F8, F14 and F18 also show a rapid convergence at first. This shows how high the convergence rate of the developed algorithm is. This is achieved by the balance of exploitation and exploration phases in the algorithm. In general, it was observed that it converges better than both algorithms.

4. Conclusion

This study was inspired by the number of update distances used in the SCA algorithm. π number as multiplier in the original algorithm used in vector $\vec{C}$. The algorithm was successful because the modified multiplier was used in both the exploitation and exploration phase. The method developed was named PIWOA (Improved Whale Optimization Algorithm Based On π Number). 23 test problems were used to compute the exploration, exploitation, and convergence curve of the improved model. PIWOA has been found to be successful in comparison with the famous meta-heuristic methods.

Acknowledgment

Thank to Dr. Seyedali Mirjalili for his scientifically motivating research and for clearly sharing them with everyone.

References

[1] Alatas, B. “ACROA: Artificial Chemical Reaction Optimization Algorithm for global optimization.” Expert Systems with Applications 38, 13170–13180, 2011.

[2] Hatamlou A. “Black hole: a new heuristic optimization approach for data clustering,” Inf Sci, 222:175–184. 2013.

[3] Huang F, Wang L, He Q. “An effective co-evolutionary differential evolution for constrained optimization,” Appl Math Computation, 186(1), 340–356, 2007.

[4] Kirkpatrick S, Gelatt CD, Vecchi MP. “Optimization by simulated annealing,” Science, 220(4598), 671–680. 1983.

[5] CernýV. “Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm,” Journal of Optimization Theory and Applications, 45(1), 41–51, 1985.

[6] Rashedi E, Nezamabadi-Pour H, Saryazdi S. “GSA: a gravitational search algorithm,” Inf Sci, 179,2232–2248,2009.

[7] Mirjalili, S., Mirjalili, S. M., & Hatamlou, “A. Multi-verse optimizer: a nature-inspired algorithm for global optimization” Neural Computing and Applications, 27(2), 495-513. 2016.

[8] Dorigo M, Birattari M, Stutzle T. “Ant colony optimization,” IEEE Comput Intell, 1(4), 28–39. 2006.

[9] Kennedy J, Eberhart R. “Particle swarm optimization,” In: Proceedings of the 1995 IEEE international conference on neural networks, Australia, 1942–1948, 1995.

[10] Basturk B, Karaboga D. “An artificial bee colony (ABC) algorithm for numeric function optimization,” In: Proceedings of the IEEE swarm intelligence symposium, Indianapolis, USA, 12–14 May 2006.

[11] Tan Y, Zhu Y. “Fireworks algorithm for optimization.” Advances in swarm intelligence, Berlin: Springer-Verlag; 355–364. 2010.

[12] Fogel D. Artificial intelligence through simulated evolution. Wiley-IEEE Press; 2009.

[13] Glover F. Tabu search –Part I. ORSA J Comput 1989; 1:190–206.

[14] Glover F. Tabu search –Part II. ORSA J Comput 1990; 2:4–32.

[15] Geem Z,W. Kim J,H., Loganathan G., “A new heuristic optimization algorithm: harmony search,” Simulation,76(2),60–68, 2001.

[16] Mirjalili S., “SCA: A Sine Cosine Algorithm for solving optimization problems,” Knowledge-Based Sys,96,120-133, 2016.

[17] Salimi H., “Stochastic fractal search: a powerful metaheuristic algorithm,” Knowledge-Based Sys,75,1-18, 2015.

[18] Eskandar H, Sadollah A, Bahreininejad A, Hamdi M., “Water cycle algorithm–A novel metaheuristic optimization method for solving constrained engineering optimization problems,” Comp & Struct, 110,151-166, 2012.

[19] Wolpert DH, Macready WG., “No free lunch theorems for optimization,” Evolut Comput, IEEE Trans, 1:67–82, 1997.

[20] Ateş, V., & Barışçı, N. “Short-Term Load Forecasting Model Using Flower Pollination Algorithm,” International Scientific and Vocational Studies Journal, 1(1), 22-29, 2017.

[21] Koç, İ. B., Al Janadi, A., & Ateş, V. “Interlock Optimization Of An Accelerator Using Genetic Algorithm,” International Scientific and Vocational Studies Journal, 1(1), 30-41. 2017.

[22] Trivedi, I. N., Pradeep, J., Narottam, J., Arvind, K., & Dilip, L. “Novel adaptive whale optimization algorithm for global optimization,” Indian Journal of Science and Technology, 9(38), 319-326, 2016.

[23] Hu, H., Bai, Y., & Xu, T., “A whale optimization algorithm with inertia weight,” WSEAS Trans. Comput, 15, 319-326, 2016.

[24] Kaveh, A, Ghazaan, M. I., “Enhanced whale optimization algorithm for sizing optimization of skeletal structures,” Mechanics Based design of structures and Machines, *45*(3), 345-362, 2017.

[25] Hu, H, Bai, Y, Xu, “Improved whale optimization algorithms based on inertia weights and theirs applications,” International journal of circuits, systems and signal processing, 11, 12-26. 2017.

[26] Ling, Y., Zhou, Y., Luo, Q. Lévy, “flight trajectory-based whale optimization algorithm for global optimization.,” IEEE access,*5*, 6168-6186, 2017.

[27] Oliva, D., El Aziz, M. A., “Hassanien, A. E. Parameter estimation of photovoltaic cells using an improved chaotic whale optimization algorithm,” Applied Energy, 200, 141-154. 2017.

[28] Wang, J., Du, P., Niu, T., Yang, W., “A novel hybrid system based on a new proposed algorithm—Multi-Objective Whale Optimization Algorithm for wind speed forecasting,” Applied energy, 208, 344-360, 2017.

[29] Abdel-Basset, M., Abdle-Fatah, L., Sangaiah, A. K. “An improved Lévy based whale optimization algorithm for bandwidth-efficient virtual machine placement in cloud computing environment,” Cluster Computing, 1-16, 2018.

[30] Kaur, G., Arora, S. “Chaotic whale optimization algorithm,” Journal of Computational Design and Engineering, 5(3), 275-284, 2018.

[31] Saidala, R. K., Devarakonda, N. “Improved whale optimization algorithm case study: clinical data of anaemic pregnant woman,” In Data engineering and intelligent computing, Springer, Singapore. 271-281, 2018.

[32] Mirjalili S, Lewis A., “The whale optimization algorithm,” Adv Eng Softw, 95:51–67, 2016.

[33] El Aziz, M. A, Ewees, A. A, Hassanien, A. E., “Multi-objective whale optimization algorithm for content-based image retrieval,” Multimedia Tools and Applications, 77(19), 26135-26172, 2018.

[34] Xiong, G., Zhang, J., Shi, D., He, Y. “Parameter extraction of solar photovoltaic models using an improved whale optimization algorithm,” Energy conversion and management, 174(1), 388-405, 2018.

[35] Abdel-Basset, M. El-Shahat, D, Sangaiah, A. K. A, “modified nature inspired meta-heuristic whale optimization algorithm for solving 0–1 knapsack problem,” International Journal of Machine Learning and Cybernetics, 10(3), 495-514, 2019.

[36] Danacı, M., & Alızada, B., “An Improvement Of Hybrid Whale Optimization Algorithm,” Euroasia Journal of Mathematics-Engineering Natural & Medical Sciences, vol.2, 60-68, 2019.

[37] Yao X, Liu Y, Lin G. “Evolutionary programming made faster,” IEEE Trans Evol Comput,3:82–102, 1999.

[38] Digalakis J, “Margaritis K. On benchmarking functions for genetic algorithms,” Int J Comput Math, 77:481–506, 2001.

[39] Molga M, Smutnicki C. Test functions for optimization needs. 2005; https://www.robertmarks.org/Classes/ENGR5358/Papers/functions.pdf

[40] Yang X-S. “Firefly algorithm, stochastic test functions and design optimization,” Int J Bio-Inspired Comput, 2(2):78–84. 2010.