International Journal of Mathematics & Computational Frontiers, Vol-1, Issue-01, pages 65-79, Jan-June 2025
https://www.ijmcf.com/

A Novel Population Segmentation-Aquila Optimizer for solving Engineering Design problems

[1,*]Megha Varshney, [2]Pravesh Kumar, [3]Pinkey Chauhan

[1,2]Rajkiya Engineering College, Bijnor, 246725, India

[3]Indra Gandhi National Open University, Delhi, 110068, India

[*]megha.math21@recb.ac.in, pkumarrecb@gmail.com, chauhanpinkey03@gmail.com

Received:20 Feb 2025**; Accepted:** 18 Mar 2025**; Published:** 2 Apr 2025

Abstract:

This research introduces a refined variant of the Aquila optimizer (AO) through a novel population segmentation technique. The new algorithm named Population Segmentation-Aquila Optimizer (PS-AO), strategically partitions search agents to achieve better exploration-exploitation trade-offs and increased solution diversity. We validate PS-AO's effectiveness through extensive testing on IEEE CEC 2019 benchmark functions and two mechanical engineering design problems. The experimental outcomes reveal that PS-AO achieves superior performance metrics in solution accuracy.

Keywords: Optimization; Nature Inspired Metaheuristic Algorithm; Aquila Optimizer; Engineering Design Problem

1. Introduction

One of the most promising fields for resolving practical optimisation issues is swarm intelligence [1]. Swarm intelligence is the ability of social organisms to work together to acquire food and create an intelligent structure [2]. By mimicking the swarming behavior of many sentient animals, including as birds, whales, wolves, ants, and bees, a variety of techniques have been devised to handle non-linear, non-convex, and discontinuous problems. Numerous examples, such as Particle Swarm Optimization (PSO) [3], Salp Swarm Algorithm (SSA) [4], Grasshopper Search Algorithm (GSA) [5], Reptile Search Algorithm (RSA) [6], Jaya Algorithm (JA) [7], Whale Optimization Algorithm (WOA) [8], Grey Wolf Optimizer

(GWO) [9], and many more, demonstrate that swarm intelligence-based algorithms can handle real-world optimization problems.

The revelation of the No Free Lunch (NFL) theorem was one of the more fascinating developments in numerical optimization [10]. This theorem states that when the execution time of an optimisation (search) method is distributed across the whole set of possible functions, all optimisation (search) methods perform similarly. In other words, the mere fact that one algorithm can handle a problem successfully does not imply that another algorithm can do it just as well. A large number of optimisation techniques, often inspired by nature, are based on this theorem [11].

The Aquila Optimizer (AO) is based on hunting behaviour of Aquila bird introduced by Abualigah et al. [12]. AO algorithm is relatively a new contribution to the family of swarm intelligence based metaheuristics which simulates four unique hunting methods of Aquila. The AO algorithm is a straightforward population-based algorithm that mimics the social and hunting behaviors of aquila birds in order to locate prey.

AO has been widely used in many settings due to its strong robustness and worldwide exploration capacity. Phase-locked loop, PID coefficients were modified using AO by Guo et al. [13] in order to lessen power fluctuations and improve the quality of the grid connection. An essential component of the PV inverter is the PLL. Hussan et al. [14] used AO to optimize the selective harmonic removal equations for the seven-level H-bridge inverter, reducing the total harmonic distortion and a number of components. The optimal minimal entropy deconvolution (MED) filter length was determined by Vashishtha et al. [15] using AO to increase recognition accuracy for diagnosing bearing problems in Francis turbines. AlRassas et al. [16] used AO to identify the optimal network parameters for the adaptive neuro-fuzzy inference system (ANFIS) network in order to increase prediction accuracy in the setting of oil production time series forecasting.

A unique Population Segmentation (PS) approach is introduced by the proposed Aquila Optimizer (AO) modification. Using PS with the Aquila Optimizer has two main advantages. Utilizing data from various solution quality levels, first improves the equilibrium between exploration and exploitation, possibly preventing premature convergence and enhancing the capacity to escape local optima [17]. Second, combining data from different regions of the search space broadens the pool of possible solutions, which is especially useful for complicated or multimodal optimization issues [18]. These enhancements allow the improved AO to efficiently converge towards high-quality solutions over a larger variety of problem types, all the while maintaining a strong search capacity. Later, IEEE CEC 2019 functions are used to test this algorithm's performance. It is then used to tackle engineering design issues, such as the design of cantilever beams and pressure vessels.

In the subsequent sections, we first present a concise introduction to AO and its background in Section 2, then describe our enhanced variant, PS-AO, and its developmental basis in Section 3. Section 4 contains numerical results for the IEEE CEC 2019 benchmark functions. The empirical validation of results, contrasting the baseline AO with our proposed version, is detailed through statistical examination in Section 5. Results from a few real-world applications are shown in Section 6. In section 7, the conclusion and future work are suggested.

2. Aquila Optimizer (AO)

The outline of the Aquila Optimizer's operation, the physical traits of the Aquila bird, its inspiration, and the mathematical model are given in this section.

2.1. Inspiration

In 2021, Abualigah et al. [12] developed the Aquila Optimizer (AO), a revolutionary bionic, gradient-free, swarm-based meta-heuristic algorithm. The design of this algorithm is mostly inspired by the Aquila, a popular prey bird in the Northern Hemisphere. Aquila catches rabbits, marmots, and other ground animals with its razor-sharp talons, powerful feet, and fast reflexes. As a result, the AO algorithm's optimization process can be divided into four distinct phases, which are as follows in summary.

2.2. Mathematical Model

Aquilas are candidate solutions in AO, and the intended prey is identified as the best solution at each stage. First, using Eq (1), the starting population of AO is generated randomly in the search space of the given issue, just like with the basic framework of other optimization paradigms.

$$x_i = rand \times (ub - lb) + lb, \quad i = 1, 2, \ldots, N$$

where Aquila's position in the population is indicated by x, a random number within the interval 0 and 1, Aquila's population size is indicated by N, and the upper and lower boundaries of the search domain are shown by ub and lb, respectively.

In order to facilitate the shift from global exploration to local exploitation, AO sets the following conditions for the switch:

$$\begin{cases} Exploration\ execution, & if\ t < \left(\frac{2}{3}\right) \times T \\ Exploitation\ execution, & otherwise \end{cases}$$

Within this bracket, t represents the ongoing iteration count, while T denotes the total iterations permitted. The algorithm’s mathematical foundation consists of four distinct computational phases.

i. Expanded exploration

Aquila uses this phase to fly high over the ground and thoroughly search the hunting area. When it detects prey, it will plunge vertically in the direction of its target. Eq. (1) simulates this behavior.

$$x_i(t+1) = x_{best}(t) \times \left(1 - \frac{t}{T}\right) + x_m(t) - x_{best}(t) \times rand \tag{1}$$

The symbol $x_i(t+1)$ reflects the i^{th} aquila’s position changes in the subsequent iteration, with x_{best} denotes the optimal solution. Additionally, $x_m(t)$ is the population's average location for all Aquilas, and it is determined as follows:

$$x_m(t) = \frac{1}{N} \sum_{i=1}^{N} x_i(t), \text{ for all } i = 1, 2, \ldots, D \tag{2}$$

where N is the population size and $x_i(t)$ is the i^{th} aquila's current position vector.

ii. Narrowed exploration

In the second phase, the Aquila gets ready to land, swoops overhead the target it has located from a high altitude and launches an assault. The mathematical expression for this behavior is as follows:

$$x_2(t+1) = x_{best}(t) \times Levy(D) + x_r(t) + (y - x) \times rand \tag{3}$$

where $x_r(t)$ represents an Aquila position randomly selected from the current population N. Levy(.) implies the Levy flight function, which is provided as follows.

$$Levy(D) = s \times \frac{u \times \sigma}{|v|^{\frac{1}{a}}}, \quad \sigma = \left(\frac{\Gamma(1+a) \times \sin\left(\frac{\pi a}{2}\right)}{\Gamma\left(\frac{(1+a)}{2}\right) \times a \times 2^{\left(\frac{a-1}{2}\right)}} \right)$$

where a is a constant value equal to 1.5, u and v are random numbers within the interval [0,1], and y and x represent the contour spiral shape during the search in Eqs. (4), (5). This contour spiral shape can be determined as follows:

$$y = r + UD_1 \cos\left(-\omega D_1 + \left(\frac{3\pi}{2}\right)\right) \tag{4}$$

$$x = r + UD_1 \sin\left(-\omega D_1 + \left(\frac{3\pi}{2}\right)\right) \tag{5}$$

During the search iterations, the control parameter fluctuates within the interval [1,20]. Two constants ω and U, are predefined: one at 0.005 and another at 0.00565. $D_1 \in \mathbf{Z}$, ranging $[1, D]$.

iii. Expanded Exploitation

The Aquila bird carefully surveys the prey area during the investigation phase before making a low, slow fall attack. Eq. (6) provides a mathematical representation of the method known as expanded exploitation, or x_3.

$$x_3(t+1) = (x_{best}(t) - x_m(t)) \times \theta - rand + ((ub - lb) \times rand + lb) \times \rho \quad (6)$$

where θ and ρ are the exploitation control coefficients set as 0.1.

iv. Narrowed Exploitation

The design of the narrowed exploitation technique is based on this hunting strategy. This case's mathematical form is provided as follows:

$$x_4(t+1) = J \times x_{best}(t) - (P_1 \times rand \times x_1(t)) - P_2 \times Levy(D) + rand \times P_1 \quad (7)$$

$$J(t) = t^{\frac{2 \times rand - 1}{(1-T)^2}} \quad (8)$$

$$P_1 = 2 \times rand - 1 \quad (9)$$

$$P_2 = 2 \times \left(1 - \frac{t}{T}\right) \quad (10)$$

where J is the quality function that balances the search strategy; P_1 is the prey's movement parameter, a random number from $[-1,1]$; and P_2 indicates the flight slope, which decreases linearly from 2 to 0, as the Aquila follows the prey from the first to the last location.

3. Proposed Algorithm

The population segmentation technique of the Aquila Optimizer (AO) aims to increase the variety of solutions and balance between exploration and exploitation. This technique divides the population into three groups based on fitness: x_{best}, x_{medium}, and x_{worst}. The sizes of these segments are λ_1, λ_2, and λ_3, respectively. The algorithm then selects vectors x_r, x_s, and x_t, from these segments; for primary computations, x_r is chosen from x_{best}. The Population Segmentation-Aquila Optimizer (PS-AO) is the name of this suggested algorithm. By using data from solutions with different quality levels, the enhanced AO may be able to prevent early convergence and increase adaptability to a variety of problem types. By dynamically changing the segments in each iteration, the approach maintains a consistent balance between using known solutions and exploring new areas of the search space, potentially leading to improved performance in a range of optimization scenarios. Fig. 1 shows the flowchart for the suggested algorithm PS-AO.

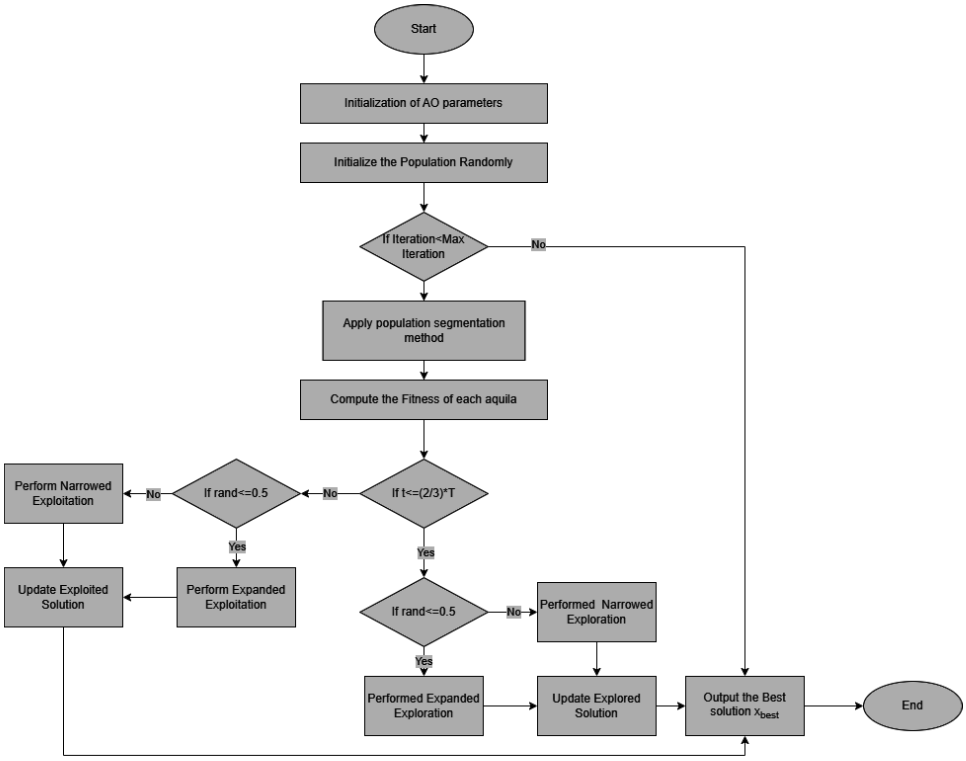

Fig. 1 The flowchart of the proposed algorithm PS-AO

4. Numerical Experiment

This section examines the performance of AO and the suggested PS-AO using the IEEE CEC 2019 benchmark functions as a baseline [19]. Ten unconstrained optimization tasks in a range of difficulty are included in this benchmark set. These numerical experiments were all carried out using MATLAB 2019a.

4.1. Benchmark Functions and Parameter Setting

In accordance with the recommendations supplied by IEEE CEC 2019, 51 runs are conducted for each benchmark function in order to observe the performance of both methods. For every variable, the search space has a range of $[-100,100]$. Table 1. presents the parameter settings of the algorithms

Table 1. Algorithms and their parameter settings	
Algorithm	**Parameters**
PS-AO	$\alpha = 0.1, \beta = 0.1$
AO [12]	$\alpha = 0.1, \beta = 0.1$
MAO [20]	$\alpha = 0.1, \beta = 0.1$
SSA [4]	$v = 0$
WOA [8]	$v = 1$

4.2 Examination of the Outcomes

In accordance with IEEE CEC 2019 reporting guidelines, we provide detailed performance analyses for the five algorithms—PS-AO, AO, MAO, WOA, and SSA—in Table 1. For every test function, the assessment metrics consist of extreme values (Best, Worst), statistical measures (Mean, STD), and absolute error numbers. Superior performance measures are indicated by bolded entries, and data regularly demonstrates PS-AO's advantages over the original AO. Algorithm comparison ratios W/L/T (Win/Loss/Tie) and processing durations are shown in the final row, where SSA exhibits the most effective computing performance.

Table 2. Statistical performance metrics (Mean, STD, Best, Worst) of AO, PS-AO, and other metaheuristic algorithms on IEEE CEC 2019 functions.

Function	PS-AO	AO	MAO	WOA	SSA
F1 Mean	**9.90E+01**	**9.89E+01**	1.23E+09	6.78E+06	7.33E+09
STD	0.00E+00	2.05E+05	7.35E+08	7.46E+06	3.48E+09
Worst	9.99E+01	5.72E+04	1.28E+09	5.55E+06	7.77E+09
Best	9.89E+01	3.90E+04	1.23E+09	6.25E+06	7.25E+09
F2 Mean	**1.95E+02**	**1.95E+02**	2.82E+04	7.66E+02	2.01E+02
STD	0.00E+00	0.00E+00	7.30E+03	8.73E+02	2.08E+-02
Worst	1.99E+02	1.82E+02	2.56E+04	6.99E+02	3.11E+02
Best	1.93E+02	1.75E+02	2.95E+04	7.88E+02	2.00E+02

F3 Mean	2.94E+02	2.82E+02	2.86E+02	2.95E+02	2.97E+02
STD	1.32E+00	1.86E+00	4.42E+-01	1.92E+00	1.77E+-15
Worst	2.99E+02	2.88E+02	2.95E+02	3.15E+02	3.25E+02
Best	2.85E+02	2.95E+02	2.82E+02	2.99E+02	2.85E+02
F4 Mean	3.44E+02	3.68E+02	2.44E+02	3.49E+02	**3.43E+01**
STD	1.37E+01	9.69E+00	2.50E+01	2.49E+01	1.07E+01
Worst	3.55E+02	3.58E+02	2.58E+02	4.52E+02	4.11E+01
Best	3.25E+02	3.25E+02	2.44E+02	3.56E+02	3.25E+01
F5 Mean	4.93E+02	4.88E+02	**3.05E+02**	4.97E+02	5.49E+02
STD	4.44E+00	1.82E-01	4.89E+01	4.59E-01	8.53E+-01
Worst	4.28E+02	4.99E+02	3.22E+02	5.11E+02	4.12E+02
Best	4.21E+02	4.01E+02	3.00E+02	4.99E+02	5.22E+02
F6 Mean	**5.92E+02**	5.89E+02	5.99E+02	5.91E+02	5.98E+02
STD	1.78E+00	1.43E+00	9.22E+-01	1.75E+00	8.53E+-01
Worst	5.99E+02	5.99E+02	6.12E+02	6.12E+02	6.12E+02
Best	5.12E+02	5.20E+02	5.12E+02	5.85E+02	5.92E+02
F7 Mean	7.16E+02	**3.10E+02**	2.22E+03	7.64E+02	4.73E+02
STD	2.56E+02	2.93E+02	2.93E+02	3.00E+02	9.77E-01
Worst	8.33E+02	6.25E+02	2.69E+03	6.55E+02	4.96E+02
Best	7.11E+02	5.36E+02	2.11E+03	7.99E+02	4.25E+02
F8 Mean	7.96E+02	8.95E+02	7.65E+02	**5.95E+02**	9.08E+02
STD	1.99E-01	3.01E-01	2.39E+-01	3.21E+-01	6.13E+-01
Worst	8.01E+02	8.99E+02	7.25E+02	6.22E+02	9.55E+02
Best	7.55E+02	8.01E+02	7.12E+02	5.69E+02	9.00E+02
F9 Mean	**8.99E+02**	9.36E+02	8.99E+03	8.98E+03	2.42E+03
STD	1.64E+-01	1.43E-01	8.68E+-01	2.01E+-01	5.96E+-01
Worst	8.99E+02	8.58E+02	8.56E+03	9.11E+03	3.11E+03
Best	8.58E+02	9.56E+02	7.25E+03	8.25E+03	2.55E+03
F10 Mean	**9.79E+02**	9.99E+02	9.86E+02	9.95E+02	2.10E+03
STD	7.69E+-01	4.63E+00	1.35E+-01	1.33E+-01	3.56E+01
Worst	9.99E+02	9.99E+02	9.25E+02	9.99E+02	3.25E+03
Best	9.65E+02	9.22E+02	8.07E+02	8.56E+02	2.00E+03
W/L/T	5/5/2	3/7/2	1/9/0	1/9/0	1/9/0
CPU Runtime	3.01E+04	3.01E+04	2.23E+04	5.11E+04	**4.32E+03**

Overall, the PS-AO algorithm performs better than AO in terms of investigating, taking advantage of, and escaping the stagnation in local optima.

4.3. Analytical statistics

This section examines the performance of AO and PS-AO using two distinct statistical techniques.

1. Wilcoxon examination
2. Friedmann Rank and Bonferroni-Dunn test

1. The Wilcoxon examination

To assess the statistical validity of the PS-AO results, the non-parametric pairwise

Wilcoxon test was utilized. The test was run with a significance level of 5% [21]. The findings are displayed in Table 5. The following standards were used to draw conclusions about the findings.

1. If the p-value is less than 0.01 then the observed difference is highly significant.
2. If the p-value is less than 0.05, then the observed difference is significant.
3. If p-value is less than 0.05, both techniques are statistically equivalent.

Results from Table 3, show that PS-AO is performing better than other algorithms.

Table 3. Wilcoxon rank sum test for the algorithms

Algorithms		ΣR^+	ΣR^-	z-value	*p*-value	Sign
PS-AO *vs*	**AO**	27	18	0.533	0.594	=
	MAO	37	18	0.969	0.333	=
	WOA	46.50	8.50	1.938	0.050	+
	SSA	37	18	0.969	0.333	=

2. Friedmann rank test and Bonferroni test

The Friedmann rank [22] is evaluated using the mean value from Table 2. From the results, the superior performance of PS-AO over AO and other metaheuristic algorithms is evident. The last two lines of the Table 4 show the CD value at $\alpha = 0.1$, and $\alpha = 0.05$ for Bonferroni Dunn CD- bar chart test. From the Fig. 2, it is clear that PS-AO has a lower bar than others. But AO is also performing well.

Table 4. Friedmann rank based on the mean value of Table 2 and CD value at the significant levels.

Algorithm	Friedmann Rank
PS-AO	**2.35**
AO	2.55
MAO	3.40
WOA	3.33
SSA	3.50
CD value at $\alpha = 0.1$	3.9346
CD value at $\alpha = 0.05$	4.1164

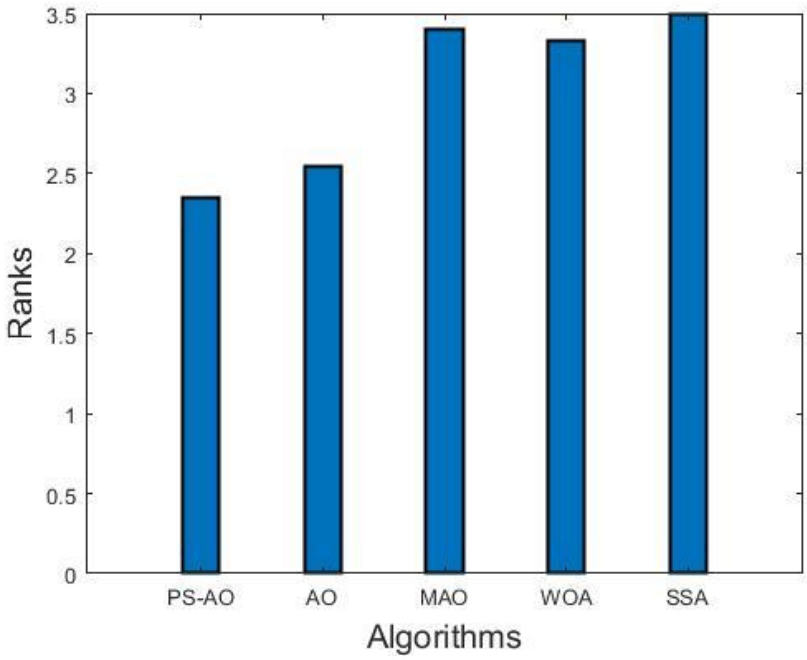

Fig. 1 Bonferroni Dunn bar chart represents the rank of the correspondence algorithm based on Friedmann rank.

5. PS-AO for Engineering Design Problems

This section uses the Cantilever Beam (CB) design, and Welded Beam (WB) design problem, to verify that PS-AO performs better when applied to real-world challenges [23]. To investigate the statistical aspects of the results, thirty separate runs of each problem were conducted.

5.1. CB design problem

The goal of the CB design challenge is to reduce a cantilever beam's weight while accounting for the vertical displacement constraint. It is necessary to optimise each of the five side length values for the five hollow square blocks. The mathematical model is described in the literature [24].

The CB design issue results are compared by four distinct MAs, including MAO, AO, SSA, and WOA, in Table 5. The outcomes show that, in comparison to other algorithms, the PS-AO, is capable of producing better results. For this reason, PS-AO is the best approach to solving the CB design issue.

Table 5. CB design results by using PS-AO, and other MAs algorithms.

Algorithm	x_1	x_2	x_3	x_4	x_5	**Optimal Weight**
PS-AO	6.1221	5.0812	4.9110	3.1223	2.1601	**1.3210**
MAO [21]	6.0172	5.3071	4.4912	3.5081	2.1499	1.3999
AO [10]	5.9134	5.4567	4.4672	3.6012	2.1034	1.3423
SSA [8]	5.9095	5.5432	4.6014	3.6534	2.4538	1.3568
WOA [22]	6.1343	5.0651	4.3430	3.8543	2.4073	1.3465

5.2. WB design problem

The objective of the WB design problem challenge is to lower the cost of producing a welded beam. The optimization parameters are clamping bar length (L), height (H), thickness (T_T), and thickness (B_B). An explanation of the mathematical model is provided in the literature [25].

Table 6 compares the outcomes of the WB design issue by four different MAs: COA, AO, SSA, and WOA. The outcomes show that AO shows better results than other MAs. But the second-best algorithm is PS-AO. So, the PS-AO is capable of producing better results.

Table 6. WB design results by using PS-AO, and other MAs algorithms.

Algorithm	H	L	T_T	B_B	**Optimal Cost**
PS-AO	0.2217	3.2432	9.0322	0.2122	1.7011
COA [21]	0.2442	3.1234	9.0344	0.2103	1.7012
AO [10]	0.1767	3.3563	9.0122	0.2134	**1.6555**
SSA [17]	0.2123	3.5211	9.0411	0.2132	1.7344
WOA [7]	0.2145	3.5461	9.0410	0.2133	1.7345

6. Conclusion and Future Scope

With the addition of a population segmentation into AO population, this paper suggests a modified form of AO that maximizes the aquila bird's capacity to search for prey. A collection of ten standard IEEE CEC 2019 benchmark problems were used to assess the suggested PS-AO algorithm's robustness. When the suggested algorithm's performance is compared to that of other metaheuristic algorithms and basic AO, it becomes clear that PS-AO is quite competitive with the other algorithms. Every analysis of the data is conducted using the standards established by IEEE CEC 2019. Based on the results of this article's investigation, it is suggested that PS-AO performs better than AO in terms of both computational time and error value. Additionally, PS-AO performs noticeably better than AO and other cutting-edge algorithms for the sample of application issues the study describes.

PS-AO may also be developed in the future to solve many optimization issues, such as integer programming, and constrained optimization problems.

References

[1] Bansal, J. C., Singh, P. K., & Pal, N. R. (Eds.). (2019). *Evolutionary and swarm intelligence algorithms* (Vol. 779). Springer.

[2] Chakraborty, A., & Kar, A. K. (2017). Swarm intelligence: A review of algorithms. In *Nature-inspired computing and optimization: Theory and applications* (pp. 475-494). Springer.

[3] Eberhart, R., & Kennedy, J. (1995). Particle swarm optimization. In *Proceedings of the IEEE international conference on neural networks* (Vol. 4, pp. 1942-1948).

[4] Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software, 114*, 163-191.

[5] Mirjalili, S. Z., Mirjalili, S., Saremi, S., Faris, H., & Aljarah, I. (2018). Grasshopper optimization algorithm for multi-objective optimization problems. *Applied Intelligence, 48*, 805-820.

[6] Abualigah, L., Abd Elaziz, M., Sumari, P., Geem, Z. W., & Gandomi, A. H. (2022). Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer. *Expert Systems with Applications, 191*, 116158.

[7] Rao, R. V. (2019). *Jaya: An advanced optimization algorithm and its engineering applications*.

[8] Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software, 95*, 51-67.

[9] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69*, 46-61.

[10] Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation, 1*(1), 67-82.

[11] Adam, S. P., Alexandropoulos, S.-A. N., Pardalos, P. M., & Vrahatis, M. N. (2019). No free lunch theorem: A review. In *Approximation and optimization: Algorithms, complexity and applications* (pp. 57-82).

[12] Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A. A., Al-Qaness, M. A. A., & Gandomi, A. H. (2021). Aquila optimizer: A novel meta-heuristic optimization algorithm. *Computers & Industrial Engineering, 157*, 107250.

[13] Aribowo, W., Supari, S., & Suprianto, B. (2022). Optimization of PID parameters for controlling DC motor based on the Aquila optimizer algorithm. *International Journal of Power Electronics and Drive Systems, 13*(1), 216.

[14] Hussan, M. R., Sarwar, M. I., Sarwar, A., Tariq, M., Ahmad, S., Mohamed, A. S. N. M., Khan, I. A., & Khan, M. M. A. (2022). Aquila optimization-based harmonic elimination in a modified H-bridge inverter. *Sustainability, 14*(2), 929.

[15] Nadimi-Shahraki, M. H., Taghian, S., Mirjalili, S., & Abualigah, L. (2022). Binary Aquila optimizer for selecting effective features from medical data: A COVID-19 case study. *Mathematics, 10*(11), 1929.

[16] AlRassas, A. M., Al-qaness, M. A. A., Ewees, A. A., Ren, S., Abd Elaziz, M., Damaševičius, R., & Krilavičius, T. (2021). Optimized ANFIS model using Aquila Optimizer for oil production forecasting. *Processes, 9*(7), 1194.

[17] Kumar, P., & Ali, M. (2023). SaMDE: A Self-Adaptive Choice of DNDE and SPIDE Algorithms with MRLDE. *Biomimetics, 8*(6), 494.

[18] Kumar, P., & Sharma, A. (2022). MRL-JAYA: A fusion of MRLDE and JAYA algorithm. *Palestine Journal of Mathematics, 11*.

[19] Liang, J.-J., Qu, B., Gong, D. W., & Yue, C. (2019). Problem definitions and evaluation criteria for the CEC 2019 special session on multimodal multiobjective optimization. *Computational Intelligence Laboratory, Zhengzhou University*, 353-370.

[20] Ali, M. H., Salawudeen, A. T., Kamel, S., Salau, H. B., Habil, M., & Shouran, M. (2022). Single-and multi-objective modified Aquila optimizer for optimal multiple renewable energy resources in distribution network. *Mathematics, 10*(12), 2129.

[21] Halim, A. H., Ismail, I., & Das, S. (2021). Performance assessment of the metaheuristic optimization algorithms: An exhaustive review. *Artificial Intelligence Review, 54*(3), 2323-2409.

[22] Eftimov, T., Korošec, P., & Koroušić Seljak, B. (2017). A novel approach to statistical comparison of meta-heuristic stochastic optimization algorithms using deep statistics. *Information Sciences, 417*, 186-215.

[23] Abualigah, L., Abd Elaziz, M., Khasawneh, A. M., Alshinwan, M., Ibrahim, R. A., Al-Qaness, M. A. A., Mirjalili, S., Sumari, P., & Gandomi, A. H. (2022). Meta-heuristic optimization algorithms for solving real-world mechanical engineering design problems: A comprehensive survey, applications, comparative analysis, and results. *Neural Computing and Applications*, 1-30.

[24] Wang, F.-Y. (1995). Optimum design of vibrating cantilevers: A classical problem revisited. *Journal of Optimization Theory and Applications, 84*, 635-652.

[25] Deb, K. (1991). Optimal design of a welded beam via genetic algorithms. *AIAA Journal, 29*(11), 2013-2015.