An Improved Whale Optimization Algorithm for Function Optimization Problems (Postprint)

Authors: Liu Liang, He Qing

Date: 2019-01-28T00:00:00+00:00

Abstract

To improve the performance of the whale optimization algorithm in solving complex function optimization problems, an improved whale optimization algorithm based on adaptive parameters and niche technology is proposed. First, an adaptive probability threshold is introduced to coordinate the algorithm' s global exploration and local exploitation capabilities. Second, adaptive position weighting is utilized to adjust the whale position update formula, enhancing the algorithm' s convergence speed and optimization accuracy. Finally, a pre-selection niche technique is employed to prevent premature convergence of the algorithm. Simulation results on twelve typical benchmark test functions demonstrate that the proposed algorithm achieves significant improvements in optimization accuracy and convergence speed compared with the comparison algorithms, proving that the proposed improvement strategies can effectively enhance the performance of the whale optimization algorithm in solving complex function optimization problems.

Full Text

Preamble

Vol. 37 No. 4
Application Research of Computers
ChinaXiv Cooperative Journal

An Improved Whale Optimization Algorithm for Solving Function Optimization Problems

Liu Liang a,b, He Qing a,b†

(a. College of Big Data & Information Engineering; b. Guizhou Provincial Key Laboratory of Public Big Data, Guizhou University, Guiyang 550025, China)

Abstract: To enhance the performance of the whale optimization algorithm in solving complex function optimization problems, this paper proposes an improved whale optimization algorithm based on adaptive parameters and niche technology. First, an adaptive probability threshold is introduced to coordinate the algorithm' s global exploration and local exploitation capabilities. Second, adaptive position weights are utilized to adjust the whale position update formula, improving the algorithm' s convergence speed and optimization precision. Finally, a preselection niche technique is employed to prevent premature convergence. Simulation results on 12 typical benchmark functions demonstrate that the improved algorithm achieves significant improvements in optimization precision and convergence speed compared with competing algorithms, proving that the proposed strategies effectively enhance the performance of the whale optimization algorithm for complex function optimization problems.

Keywords: function optimization; whale optimization algorithm; adaptive parameters; niche

Chinese Library Classification: TP301.6
doi: 10.19734/j.issn.1001-3695.2018.11.0726

0 Introduction

The Whale Optimization Algorithm (WOA) [?], proposed by Mirjalili et al. in 2016, is a novel swarm intelligence optimization algorithm that simulates the hunting behavior of humpback whales. The core idea of this algorithm is to solve target problems by imitating the predatory behavior of whales [?]. Experimental results have demonstrated that WOA outperforms classical algorithms such as Particle Swarm Optimization (PSO) [?] and Gravitational Search Algorithm (GSA) [?] in terms of both convergence speed and optimization precision. WOA has been successfully applied to water resource optimization allocation [?], optimal control [?], and feature selection [?]. However, like other swarm optimization algorithms, traditional WOA still suffers from premature convergence, slow convergence speed, and inability to find the global optimum.

To improve the convergence speed and optimization precision of WOA, scholars have proposed various improvements from different perspectives in recent years. For instance, Abdel-Basset et al. [?] utilized Lévy flights and logistic chaotic mapping to replace the coefficient vectors and determine the switching probability p in WOA, proposing an improved WOA algorithm and demonstrating its effectiveness and superiority through experiments. Sayed et al. [?] introduced chaotic search into the iterative process of WOA, proposing a CWOA algorithm that uses chaotic maps to define parameters such as A, C, l, and p. Comparisons with WOA and ten other optimization algorithms proved that this method significantly improves WOA performance. Guo Zhenzhou et al. [?] proposed a WOAMC algorithm by applying Cauchy mutation to whale positions and introducing adaptive weight coefficients, demonstrating through experiments that this algorithm outperforms traditional WOA in convergence precision and stability. Long Wen et al. [?] proposed an IWOA algorithm that coordinates exploration and exploitation capabilities through a nonlinear convergence factor update formula and performs diversity mutation operations on the current best individual, with experiments showing significantly improved optimization precision and convergence speed. Wang Jianhao et al. [?] proposed an improved algorithm based on chaotic opposition-based learning and nonlinear chaotic perturbation, testing multiple benchmark functions to prove its enhanced performance compared with traditional WOA.

These studies have yielded many advanced results addressing WOA' s limitations. However, further research is needed to enhance the algorithm' s optimization precision and convergence speed. This paper first modifies the probability threshold for selecting whale predation strategies in WOA by replacing the fixed threshold with an adaptive parameter to balance global exploration and local exploitation during iteration. Second, the adaptive parameter is used as a weight coefficient to adjust the whale position update formula, improving optimization precision and convergence speed. Finally, a preselection mechanism-based niche algorithm is integrated to maintain population diversity while ensuring whales move toward the "prey," preventing the algorithm from falling into local optima. Simulation results on 12 benchmark functions demonstrate that these three improvement strategies significantly enhance optimization precision and convergence speed.

---

1 Whale Optimization Algorithm

Compared with other whale species, humpback whales possess a unique hunting method known as the bubble-net feeding strategy, as illustrated in Figure 1. The WOA algorithm is a novel bio-inspired swarm intelligence algorithm derived from this behavior. Based on the characteristics of humpback whales' bubble-net feeding, WOA can be divided into three distinct phases: encircling prey, bubble-net attacking, and searching for prey.

[Figure 1: see original paper]

1.1 Encircling Prey

During predation, whales must first locate and then encircle their prey. In practical optimization problems, the position of prey in the search space is typically unknown. Therefore, WOA assumes that the current best solution in the population is the target prey. After identifying the prey, other whales in the population update their positions based on the prey' s current location as follows [?]:

$$\vec{D} = |\vec{C} \cdot \vec{X}^*(t) - \vec{X}(t)|$$

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

where t represents the current iteration number, $\vec{X}^*$ is the position of the best solution in the current population, $\vec{X}$ represents the current position of a whale, $\vec{D}$ denotes the encircling step size, and $\vec{A}$ and $\vec{C}$ are coefficient vectors defined as:

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a}$$

$$\vec{C} = 2 \cdot \vec{r}_2$$

where r_1 and r_2 are random numbers in $[0, 1]$, and $\vec{a}$ is a control parameter whose value linearly decreases from 2 to 0 with increasing iterations, expressed as:

$$a = 2 - 2 \cdot \frac{t}{\text{Max\_iter}}$$

where Max_iter is the maximum number of iterations.

1.2 Bubble-Net Attacking

Humpback whales' bubble-net feeding behavior involves moving toward prey along a spiral path within a shrinking encirclement. Therefore, WOA incorporates two strategies to simulate this unique behavior: shrinking encirclement mechanism and spiral position updating.

(a) Shrinking Encirclement Mechanism: This is achieved by decreasing the value of a in Eq. (3). As shown in Eq. (3), the value of $\vec{A}$ decreases with a. When a linearly decreases from 2 to 0, the range of $\vec{A}$ becomes $[-a, a]$. Consequently, when $\vec{A}$ takes values within $[-1, 1]$, the updated whale position will necessarily lie between its original position and the prey, causing each whale to move closer to the prey and complete the encirclement.

(b) Spiral Position Updating: This strategy first calculates the distance between the whale and the prey, then creates a spiral equation between them to mimic the spiral motion of humpback whales, as shown in Eq. (6) [**?**]:

$$\vec{X}(t+1) = \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t)$$

where $\vec{D}' = |\vec{X}^*(t) - \vec{X}(t)|$ represents the distance between the whale and the prey, b is a constant defining the spiral shape (set to $b = 1$ in this paper), and l is a random number in $[-1, 1]$.

Since whales must simultaneously shrink the encirclement and move along a spiral path toward the prey, WOA assumes that the probability of selecting either strategy during hunting is 0.5. The mathematical model is:

$$\vec{X}(t+1) = \begin{cases} \vec{X}^*(t) - \vec{A} \cdot \vec{D} & \text{if } p < 0.5 \\ \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) & \text{if } p \geq 0.5 \end{cases}$$

where p is a random number in $[0, 1]$.

1.3 Searching for Prey

Contrary to the shrinking encirclement mechanism, during the prey search phase, when $|\vec{A}| > 1$, whales search for prey randomly through each other' s positions. Instead of selecting the prey to update their positions, a random individual from the population is chosen to replace the original prey' s role, forcing whales away from the prey' s location to enhance WOA' s global search capability. The mathematical model is:

$$\vec{D} = |\vec{C} \cdot \vec{X}_{\text{rand}} - \vec{X}|$$

$$\vec{X}(t+1) = \vec{X}_{\text{rand}} - \vec{A} \cdot \vec{D}$$

where $\vec{X}_{\text{rand}}$ is a position vector of a randomly selected whale from the current population.

---

2 Improved Whale Optimization Algorithm Based on Adaptive Parameters and Niche

To enhance the performance of WOA in solving complex function optimization problems, this paper combines three improvement strategies to propose an Adaptive Parameter and Niche-based Whale Optimization Algorithm (APN-WOA).

2.1 Adaptive Probability Threshold

During the bubble-net attacking phase, WOA sets the probability of selecting either strategy to 50% (threshold = 0.5) to simulate the simultaneous execution of shrinking encirclement and spiral position updating, as shown in Eq. (7). By comparing a randomly generated p value in $[0, 1]$ with this threshold, the feeding strategy is selected. However, this equal-probability strategy selection may prevent whales from choosing appropriate predation strategies for the current population, leading to slow convergence and local optima problems. Therefore, this paper introduces an adaptive parameter to replace the fixed probability threshold. This adaptive threshold varies in $[0, 1]$ during iterations, enabling whales to have a higher probability of selecting appropriate predation strategies at different stages, thereby coordinating global exploration and local exploitation capabilities and improving convergence speed. The mathematical expression is:

$$\text{Adaptive\_p} = \lambda \cdot \left(\frac{t}{\text{Max\_iter}}\right)^2 + \mu \cdot \frac{t}{\text{Max\_iter}} + \lambda + \mu$$

where t is the current iteration number, Max_iter is the maximum iteration number, and λ, μ are control parameters with $\lambda = 3$ and $\mu = 2$ in this paper. Consequently, Eq. (7) can be rewritten as:

$$\vec{X}(t+1) = \begin{cases} \vec{X}^*(t) - \vec{A} \cdot \vec{D} & \text{if } p < \text{Adaptive\_p} \\ \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) & \text{if } p \geq \text{Adaptive\_p} \end{cases}$$

As shown in Eq. (11), during the early iterations, the adaptive threshold is larger, giving whales a higher probability of selecting the shrinking encirclement mechanism. In later iterations, the threshold becomes smaller, increasing the probability of selecting spiral position updating. Thus, whales transition from simultaneous strategy execution to first shrinking encirclement then spiral updating, enabling them to approach prey faster and improving algorithm convergence speed.

2.2 Adaptive Position Weight

From Eqs. (2), (7), and (9), the primary factors affecting whale position updates in WOA are the prey position vector $\vec{X}^*$ and the randomly selected whale position vector $\vec{X}_{\text{rand}}$. However, WOA does not consider that the guiding force of prey on whales may vary during iterations, nor that the relationship between randomly selected whales and prey differs across stages. Inspired by [**?**, **?**], this paper uses the adaptive parameter as a weight coefficient to adjust WOA' s position update formula as follows:

$$\omega = \lambda \cdot \left(\frac{t}{\text{Max\_iter}}\right)^2 + \mu \cdot \frac{t}{\text{Max\_iter}} + \lambda + \mu$$

where $\lambda = 3$ and $\mu = 2$, with ω taking values in $[0, 1]$. As ω increases with iterations, it indicates that the selected prey (the current global optimum) becomes more attractive to whales as its fitness improves. Similarly, the credibility of information conveyed by randomly selected whales increases during iterations. Therefore, in Eqs. (13) and (14), the weight coefficient increases with iterations, enabling whales to locate prey more accurately based on adaptive weight changes, thereby improving convergence speed and optimization precision. However, during later iterations when spiral position updating occurs, whales approach the prey, requiring a smaller weight coefficient as shown in Eq. (15) to better search for potentially superior solutions around the prey, thus enhancing local exploitation capability.

2.3 Preselection Niche Technology

In biology, a niche refers to a specific living environment. The earliest application of niche concepts to optimization involved combining them with genetic algorithms, enabling individuals to evolve within specific environments to avoid local optima. Current niche implementation methods include preselection-based, crowding-based, and sharing function-based approaches. This paper combines preselection-based niche concepts with WOA to maintain population diversity and enhance global search capability.

The main idea of preselection-based niche technology is that offspring can only replace their parents and be inherited by the next generation when their fitness exceeds that of their parents. Since offspring and parents have similar structures, only similar individuals are replaced, maintaining population diversity and creating a niche evolutionary environment [?]. Therefore, this paper stores the positions of each whale before and after iteration in memory matrices $\vec{M}$ and $\vec{N}$, respectively. If the fitness of the whale represented by the i-th row of $\vec{N}$ is better than that of the i-th row of $\vec{M}$, the i-th row of $\vec{M}$ is replaced by the i-th row of $\vec{N}$. In other words, if the position-updated whale has better fitness, it is retained; otherwise, the whale returns to its original position, preventing all whales from clustering at a local optimum and maintaining population diversity to improve global search capability.

The overall APN-WOA algorithm flow is shown in Figure 2 [Figure 2: see original paper].

---

3 Experimental Simulation and Analysis

All algorithm simulations were implemented on a computer with an Intel(R) Core(TM) i5-6500 CPU at 3.2 GHz, 8 GB RAM, and Windows 7 (64-bit) operating system, using MATLAB R2015b. To verify the effectiveness of the proposed APN-WOA algorithm, 12 typical benchmark functions were employed for testing, as listed in Table 1 , where F1–F6 are continuous unimodal functions and F7–F12 are complex nonlinear multimodal functions.

The algorithm testing 主要包括以下三个部分: (a) Under the same population size and iteration number conditions, compare the convergence speed and optimization precision of the improved algorithm with traditional WOA and Grey Wolf Optimizer (GWO) [?] on benchmark functions to prove the effectiveness of APN-WOA; (b) Conduct independent tests on the improved algorithm under different strategies to analyze the impact of different improvement strategies on algorithm performance; (c) Compare with other improved WOA algorithms from references to demonstrate that APN-WOA remains competitive against the latest improved WOA algorithms.

[Figure 2: see original paper]

3.1 Algorithm Performance Testing

The performance of APN-WOA was tested in search spaces of different dimensions (30/200/500). The population size was set to 30, and the maximum iteration number was 500. Three algorithms—GWO, WOA, and APN-WOA—were independently run 30 times. Algorithm performance differences were measured from two aspects: mean and standard deviation of the best solutions, with some data from [?] referenced for comparison. The test results are shown in Table 2 ("—" indicates data not provided in the reference). To more intuitively reflect APN-WOA' s performance, Figures 3 [Figure 3: see original paper]–5 [Figure 5: see original paper] present the optimization convergence curves for the 12 benchmark functions in 30-dimensional, 200-dimensional, and 500-dimensional search spaces.

Table 2 shows that APN-WOA achieves significantly improved optimization precision compared with traditional WOA for all 12 benchmark functions across different dimensions. It obtains theoretical optimal values for functions F1, F3, F8, and F10, and achieves standard deviations of 0 for functions F2, F4, and F9, approaching their theoretical values. Compared with GWO, APN-WOA' s optimization precision is almost entirely superior, only slightly inferior to GWO on function F5 at 30 dimensions and function F12 at 200 dimensions, but remaining in the same order of magnitude. As shown in Figures 3(e) and (l), when APN-WOA and GWO achieve the same order of magnitude in optimization precision, APN-WOA' s convergence speed is significantly faster. Moreover, comparing convergence curves across different dimensions reveals that for all 12 benchmark functions, APN-WOA' s convergence speed is substantially improved over both traditional WOA and GWO in 30-dimensional, 200-dimensional, and 500-dimensional search spaces, with the most significant improvements for functions F1, F2, F3, F4, F6, F8, F9, and F10. Compared with MS-WOA, APN-WOA is only slightly inferior on functions F11 and F12 but in the same order of magnitude, while significantly outperforming MS-WOA in both convergence speed and optimization precision for other functions. Even when both algorithms achieve optimal solutions for functions F1, F8, and F10, Figures 3 and 4 show that APN-WOA' s convergence speed remains markedly superior.

3.2 Impact of Improvement Strategies on Algorithm Performance

To compare the impact of different improvement strategies on APN-WOA' s performance, the same experimental parameters as in Section 3.1 were used to optimize the 12 benchmark functions in Table 1. WOA with only the adaptive probability threshold strategy is denoted as WOA-1, WOA with only the adaptive position weight strategy as WOA-2, and WOA with only the niche technology as WOA-3. The test results are shown in Table 3 .

Table 3 indicates that the adaptive probability threshold and niche strategies provide limited improvement in optimization precision, while the adaptive position weight strategy effectively enhances algorithm performance. However, the performance improvement from using adaptive position weight alone still shows a considerable gap compared with APN-WOA combining all three strategies on most benchmark functions. This demonstrates that while individual adaptive probability threshold and niche strategies cannot significantly improve optimization precision, they effectively balance global exploration and local exploitation capabilities and maintain population diversity to avoid local optima. Therefore, APN-WOA integrates the advantages of all three strategies, achieving faster convergence speed when obtaining the same order of magnitude of optimization precision, and significantly higher optimization precision than single-strategy WOA algorithms for most test functions, proving the rationality and effectiveness of employing three improvement strategies.

3.3 Performance Comparison with Other Improved WOA Algorithms

To compare APN-WOA' s performance with other improved algorithms, the population size was set to 30, maximum iterations to 500, and benchmark functions F1, F2, F5, F8, F9, F10, F11 were tested with 30 independent runs. Data from [**?**, **?**] were referenced to compare mean and standard deviation of optimal solutions, with results shown in Table 4 . Using the same population size and iteration number, benchmark functions F1–F10 were tested with 50 independent runs, with data from [**?**] referenced for comparison with the WOAMC algorithm, as shown in Table 5 .

Tables 4 and 5 show that when IWOA and CWOA achieve theoretical optimal solutions for test functions, APN-WOA can also obtain optimal solutions for corresponding functions. APN-WOA is only slightly inferior to IWOA on function F5 but outperforms both improved algorithms from the references on all other test functions. Moreover, APN-WOA achieves standard deviations of 0 for functions F2 and F9, approaching theoretical values, with mean optimal solution values for function F2 several orders of magnitude better than the other two algorithms in Table 3. Compared with the WOAMC algorithm from [**?**], APN-WOA also achieves theoretical optimal solutions for functions F1, F3, F8, and F10, with standard deviations of 0 for functions F2, F4, and F9, approaching theoretical values. The mean optimal solution values for functions F2 and F4 are several orders of magnitude better than those of the WOAMC algorithm.

In summary, the proposed APN-WOA algorithm not only significantly improves optimization precision and convergence speed compared with traditional WOA but also maintains clear advantages over current state-of-the-art improved WOA algorithms.

4 Conclusion

As a novel heuristic optimization algorithm, WOA shares similar limitations with other metaheuristic algorithms, including slow convergence and susceptibility to local optima when solving complex function optimization problems. Considering the impact of whale predation strategy selection and position updates on algorithm performance during iterations, this paper employs three improvement strategies: adaptive probability threshold, adaptive position weight, and preselection niche technology. Simulation results on 12 benchmark functions demonstrate substantial improvements in optimization precision and convergence speed, with clear advantages over other improved algorithms, proving that the proposed strategies enhance algorithm performance for complex function optimization problems. Future research will focus on applying the improved algorithm to constrained optimization problems and complex real-world engineering applications.

References

[1] Mirjalili S, Lewis A. The whale optimization algorithm [J]. Advances in Engineering Software, 2016, 95: 51-67.

[2] Zhong Minghui, Long Wen. Whale optimization algorithm based on stochastic adjustment control parameter [J]. Science Technology and Engineering, 2017, 17(12): 68-73.

[3] Kennedy J, Eberhart R. Particle swarm optimization [C]// Proc of IEEE International Conference on Neural Networks. Piscataway: IEEE Press, 1995: 1942-1948.

[4] Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: a gravitational search algorithm [J]. Information Sciences, 2009, 179(13): 2232-2248.

[5] Sha Jinxia. Application of ameliorative whale optimization algorithm to optimal allocation of multi-objective water resources [J]. Water Resources and Hydropower Engineering, 2018, 49(4): 18-26.

[6] Mehne H H, Mirjalili S. A parallel numerical method for solving optimal control problems based on whale optimization algorithm [J]. Knowledge-Based Systems, 2018, 151: 114-123.

[7] Mafarja M, Mirjalili S. Whale optimization approaches for wrapper feature selection [J]. Applied Soft Computing, 2018, 62: 441-453.

[8] 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 [EB/OL]. (2018-01-24) [2018-10-25]. https://doi.org/10.1007/s10586-018-1769-z.

[9] Sayed G I, Darwish A, Hassanien A E. A new chaotic whale optimization algorithm for features selection [J]. Journal of Classification, 2018, 35(2): 300-344.

[10] Guo Zhenzhou, Wang Ping, Ma Yunfeng, et al. Whale optimization algorithm based on adaptive weight and cauchy mutation [J]. Microelectronics & Computer, 2017, 34(9): 20-25.

[11] Long Wen, Cai Shaohong, Jiao Jianjun, et al. Improved whale optimization algorithm for large scale optimization problems [J]. Systems Engineering-Theory & Practice, 2017, 37(11): 2983-2994.

[12] Wang Jianhao, Zhang Liang, Shi Chao, et al. Whale optimization algorithm based on chaotic search strategy [EB/OL]. (2018-08-16) [2018-10-25]. https://doi.org/10.13195/j.kzyjc.2018.0098.

[13] Sun Yongjun, Wang Xilu, Chen Yahuan, et al. A modified whale optimization algorithm for large-scale global optimization problems [J]. Expert Systems With Applications, 2018, 114: 563-577.

[14] Mafarja M M, Mirjalili S. Hybrid whale optimization algorithm with simulated annealing for feature selection [J]. Neurocomputing, 2017, 260: 302-312.

[15] Dong Wenyong, Kang Lanlan, Liu Yuhang, et al. Opposition-based particle swarm optimization with adaptive elite mutation and nonlinear inertia weight [J]. Journal on Communications, 2016, 37(12): 1-10.

[16] He Qing, Wei Kangyuan, Xu Qinshuai. Mixed strategy based improved whale optimization algorithm [J/OL]. Application Research of Computers, 2019, 36(12). (2018-09-30) [2018-10-25]. http://kns.cnki.net/kcms/detail/51.1196.TP.20180929.1332.012.html.

[17] Zheng Min, Gao Junbo. Improved niche genetic algorithm for multimodal optimization [J]. Computer Systems & Applications, 2014, 23(10): 101-106.

[18] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer [J]. Advances in Engineering Software, 2014, 69(3): 46-61.

Note: Figure translations are in progress. See original paper for figures.