SEMARAK ILMU PUBLISHING

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943


# Self-Adaptive Two-Stage Anchovy-Inspired Filter Algorithm for Global Optimization

Azrul Mahfurdz[1], Muhammad Muizz Muhammad Nawawi[1], Sunardi Sasmowiyono[2]

[1] Department of Electrical Engineering, Politeknik Sultan Idris Shah, Sungai Lang, 45100 Sungai Air Tawar Selangor, Malaysia
[2] Faculty of Industrial Technology, Department of Electrical Engineering, Universitas Ahmad Dahlan, Yogyakarta 55166, Indonesia

**ARTICLE INFO**

***Article history:***
Received 5 June 2026
Received in revised form 5 August 2026
Accepted 10 August 2026
Available online 17 August 2026

***Keywords:***

Anchovy Filter Algorithm (AFA); self-adaptive optimization; swarm intelligence, Metaheuristic algorithms, two-stage filtering

**ABSTRACT**

This study aims to enhance the Anchovy Filter Algorithm (AFA), a bio-inspired metaheuristic optimization algorithm, by introducing self-adaptive parameter control and a two-stage filtering strategy to improve convergence efficiency and solution quality across various benchmark functions. A self-adaptive mechanism was incorporated into AFA to dynamically adjust parameters such as step size, weighting coefficients, and noise intensity, allowing the algorithm to automatically balance exploration and exploitation. In addition, two-stage filtering was used to refine the candidate solutions. The proposed Self-Adapting AFA was evaluated against the static AFA variant using six standard benchmark functions namely Sphere, Rosenbrock, Schwefel 1.2, Rastrigin, Griewank and Ackley. Statistical performance metrics (mean, standard deviation, best, and worst fitness values) and Wilcoxon signed-rank tests were employed for comparative analysis. The experimental results demonstrated that the Self-Adaptive AFA outperformed the static AFA in most test functions, particularly for unimodal functions such as Sphere and Ackley, achieving significantly lower mean and best fitness values with faster convergence. On multimodal functions like Rosenbrock and Rastrigin, the self-adaptive variant maintained competitive performance while ensuring better stability. The Wilcoxon test results further confirm the statistical significance of the improvements. The combination of self-adaptive parameter control and two-stage filtering significantly improves the robustness, convergence speed, and accuracy of AFA solutions. The proposed Self-Adaptive AFA provides a promising optimization framework that can be extended to real-world engineering and data-driven applications.

## 1. Introduction

Over the past decade, algorithms that emerged from observations of nature such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE) and Artificial Fish Schooling Algorithm (AFSA) have been widely used across various fields including engineering, machine learning, operations and many others. These algorithms are inspired by biological or natural phenomena to balance exploration and exploitation in the search process.

In this study, we propose the Anchovy-inspired Filter Algorithm (AFA), a novel bio-inspired optimization algorithm that simulates the collective filtering and foraging behavior of anchovies in plankton-rich environments [1]. This algorithm introduces a unique filtering mechanism where

multiple candidate solutions are generated and evaluated, while only the most promising solutions are retained for ranking updates. Through this method, it improves the balance between exploration and exploitation, enabling the AFA method to effectively tackle complex optimization problems. This filtering process enhances convergence speed and prevents premature stagnation. Two variants are commonly used: a single-stage filter where the best candidate is selected directly, and a two-stage filter where potential candidates undergo a selection process that is refined through local perturbations. Early AFA methods relied on fixed control parameters such as step size, mixing factor, and noise coefficient. These parameters were usually tuned manually and remained constant throughout the search process.

Incorrect in choosing and using the wrong values can lead to poor performance across different optimization problems. To overcome this problem, self-adaptive parameter embedding has become a popular solution and is receiving increasing attention in evolutionary computation. In this strategy, each individual in the population carries not only its position but also its control parameters, which evolve along with the search. Successful individuals propagate both their solutions and parameter configurations, leading to dynamic adjustment of the search behavior without external tuning. This approach has been successfully applied in algorithms such as Self-Adaptive Differentiation Evolution (jDE). Benchmarking tests have used six functions (Sphere, Rosenbrock, Schwefel 1.2, Rastrigin, Griewank, and Ackley) with various local minima functions to evaluate convergence behavior, robustness, and statistical significance.

## 2. Literature Review

Swarm intelligence (SI) and evolutionary computation have been chosen as very popular methods in helping to solve optimization problems involving large data sets and complex problems. Classical metaheuristic methods such as Genetic Algorithm (GA) [2,3], Particle Swarm Optimization (PSO) [4,5], Differential Evolution (DE) [6], Artificial Fish Swarm Algorithm (AFSA) [7], and Fish School Search (FSS) [8] have been widely studied and successfully applied across various fields of learning and engineering, machine learning, and machine image processing. These methods have in common the ability to mimic natural or biological behavior to balance exploration and exploitation in high-dimensional search spaces [9,10].

The critical factors that affect the performance of these algorithms are the choice of control parameters, such as step size, mutation rate, inertia weight and learning factor [11]. Typically, the parameters are set using static values specified by the user. However, based on previous studies, the choice of fixed parameters can lead to premature convergence or slow progress depending on the objective function landscape [12,13]. Therefore, fixed parameters and parameter control have become a major research topic in metaheuristics. One of the earliest taxonomies was established by Eiben *et al.,* [14], who distinguished between parameter tuning (offline adjustment before execution) and parameter control (online adjustment during the search). Later, Lobo *et al.,* [15] and Karafotias *et al.,* [16] provided extended and comprehensive reviews that highlighted the limitations of static parameter settings.

To overcome this problem, several solutions and self-adaptation have been proposed. Adaptive control mechanisms are based on feedback from the search process, often relying on deterministic or probabilistic rules [17,18]. In contrast, self-adaptive methods incorporate parameters into individual representations, allowing them to evolve along with candidate solutions [19]. Successful individuals not only propagate solution features but also transmit their parameter configurations, effectively automating the parameter tuning process [20]. This concept has been very influential in DE research, where variants such as jDE [21], Adaptive Differential Evolution with Optional External

Archive (JADE) [22], and DE based on success data information [23] have shown significant improvements over static parameters. The results of the jDE method, Brest *et al.,* [21] show that embedding the scale factor (F) and crossover rate (CR) into individuals allows the algorithm to adapt dynamically without user intervention, achieving good results across a variety of benchmarks.

Several studies have proposed modified Differential Evolution (DE) frameworks that employ ensemble mutation strategies together with self-adaptive parameters [24]. Subsequently, adaptive parameter control combined with external archive mechanisms was introduced to maintain population diversity [22]. Later, success-history based adaptive parameter adaptation was proposed, where parameter values are updated according to the success history of previous generations [23]. Similar adaptive concepts have also been applied to other metaheuristics, such as the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [25]. Furthermore, adaptive and self-adaptive strategies have been investigated in swarm intelligence and other evolutionary algorithms. For example, adaptive learning approaches have been incorporated into Particle Swarm Optimization (PSO) [26], while Artificial Fish Swarm Algorithm (AFSA) and Fish School Search (FSS) have shown limited but promising developments in this area [27,28]. In addition, hybridization and opposition-based learning strategies have been explored to accelerate convergence in self-adaptive DE variants [29] ertheless, most studies on adaptive parameter mechanisms have focused on Differential Evolution, leaving many algorithms inspired by other optimization paradigms relatively underexplored [30].

From this review, it is evident that while parameter control has been studied extensively in DE, there remains a significant research gap in extending self-adaptive parameter embedding to newer swarm-inspired algorithms. In particular, filter-based algorithms such as the Anchovy-inspired Filter Algorithm (AFA) have not been explored under adaptive frameworks. This motivates the present study, which introduces self-adaptive parameter embedding into AFA and compares its performance against static-parameter variants under one-stage and two-stage filtering strategies.

## 3. Self-Adaptive Two-Stage Anchovy-Inspired Filter Algorithm

Inspired by the natural feeding behavior of anchovies, we propose an enhanced bio-inspired optimization method. In nature, anchovies feed by filtering plankton suspended in the water. A group of anchovies swims collectively while each fish continuously filters the water, retaining edible particles and discarding the rest. This dual-stage filtration ensures efficient exploitation of nearby food sources while maintaining collective exploration of new feeding grounds. In Self-Adaptive AFA, this process is computationally modeled as follows:

- **Candidate generation (Stage 1 filtering – intake):**
  Each anchovy samples multiple candidate solutions around its position, analogous to filtering water for plankton.

- **Selective filtering (Stage 1 retention):**
  From the generated candidates, only the best per individual is retained, mimicking the selective intake of nutritious plankton.

- **Refinement filtering (Stage 2 Refinement Search):**
  The second stage of filtering refines the best solutions from Stage 1. A group of high-performing candidates undergoes Gaussian-based local perturbation, allowing for fine-tuning of potential regions.

- **Self-Adaptive parameter control:**
  Unlike fixed parameter methods, each anchovy potentially carries its own adaptive parameters (e.g., weighting coefficients & step size). These conditions evolve dynamically during the search, allowing the algorithm to automatically balance exploration and exploitation.

- **Schooling movement (collective exploration):**
  Anchovies adjust their positions towards a weighted combination of their best candidate and the global best, ensuring both local refinement and population-wide exploration.

- **Dynamic filter size:**
  The filtering radius adaptively decreases across iterations, encouraging broad exploration in early stages and focused exploitation in later stages.

- **Global best tracking**
  Update the recorded global best value (but do not insert it into population):

- **Self-adaptive parameter mutation**
  For each parameter of each individual (independently)

- **Termination**
  The adaptive Anchovy Filter Algorithm (AFA) improves upon the traditional filtering process by incorporating adaptive parameters $\alpha$ and $\beta$ in each individual. The optimization process needs to follow several steps, which are represented as mathematical equations.

**a. Initialization (School Formation)**

The initial population is randomly generated within the search space:

$$x^0 \sim U\ (lb, ub)\ , i = 1,2, \dots, N \qquad (1)$$

where N is the school size (population), and [*lb*, *ub*] defines the ocean boundaries (search space). This corresponds to a school of anchovies initially scattered across the sea.

Fitness Evaluation (Plankton Quality)
The nutritional value of plankton is represented by the fitness function:

**b. Fitness Evaluation (Plankton Quality)**

The nutritional value of plankton is represented by the fitness function:

$$f(x_i) = objective\ function\ (\ x_i) \qquad (2)$$

where better fitness values correspond to more nutritious plankton captured by an anchovy.

**c. Stage-1 Filtering (First Candidate Generation)**

The first candidate is generated using a dynamic filter:

$$x_{i,j}^{cand1}(t) = x_i\,(t) + \delta^{(1)}(t)\,.\; r_{i,j}\,, r_{i,j} \sim U(-1,1) \tag{3}$$

$$x_i(t) = \begin{cases} x_i^{cand1}(t), & f(x_i^{cand1}(t)) < f(x_i(t)) \\ x_i(t), & Otherwise \end{cases} \tag{4}$$

**d. Stage-2 Filtering (Second Candidate Generation)**

$$x_{i,j}^{cand2}(t) = x_i\,(t) + \delta^{(2)}(t)\,.\; r_{i,j}\,, r_{i,j} \sim U(-1,1) \tag{5}$$

$$x_i(t) = \begin{cases} x_i^{cand2}(t), & f(x_i^{cand2}(t)) < f(x_i^{(1)}(t)) \\ x_i^{(1)}(t), & Otherwise \end{cases} \tag{6}$$

**e. Global Best Update**

$$g(t) = \arg min_{\,i}\, f\big(\, x_i^{best}\,(t)\,\big) \tag{7}$$

**f. Position Update**

$$x_i(t+1) = x_i^{best}\,(t) + \alpha\,.\Big(g(t) - x_i^{best}(t)\Big) + \beta.r_i \;\; r_i \sim U(-1,1) \tag{8}$$

**g. Dynamic Filter Adaptation**

$$\delta^s(t) = {\delta^s}_{max} - \frac{t}{T}\,({\delta^s}_{max} \;-\; {\delta^s}_{min}\,), s = 1,2 \tag{9}$$

**h. Self-Adaptive Parameter control**

The parameters α and β are updated adaptively:

$$\alpha(t+1) = \;\alpha(t)\,.\Big(1 - \frac{t}{T}\Big) \tag{10a}$$

$$\beta(t+1) = \;\beta(t)\,.\Big(\frac{t}{T}\Big) \tag{10b}$$

Through this approach, this paper examines the performance of AFA with adaptive parameter embedding compared to its static parameter counterpart under one-stage and two-stage filtering methods.

## 2. Methodology

Six benchmark functions were employed for the simulations and tested in a 30-dimensional space, consisting of Sphere, Rosenbrock and Schwefel 1.2 (unimodal), and Rastrigin, Griewank and Ackley (multimodal with numerous local minima), which are widely used standard test functions in evolutionary computation studies [12,28]. Table 1 provides the corresponding search domain,

initialization range, and global optimum meanwhile Table 2 showed parameter setting for AFA stage 1 filter, AFA stage 2 filter, Self-Adaptive AFA stage 1 filter and Self-Adaptive AFA stage 2 filter.

$$F_{Sphere}\ (x) = \sum_{i=1}^{n} x_i^2 \tag{11}$$

$$F_{Rosenbrock}\ (x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x^2) + (x_i - 1)^2 \tag{12}$$

$$F_{Schwefel\ 1.2}\ (x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2 \tag{13}$$

$$F_{Rastrigin}\ (x) = 10n + \sum_{i=1}^{n} [x_i^2 - 10\cos\ (2\pi x_i)] \tag{14}$$

$$F_{Griewank}\ (x) = 1 + \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} cos\ \left(\frac{x_i}{\sqrt{i}}\right) \tag{15}$$

$$F_{Ackley}\ (x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right), \exp\ \left(\frac{1}{n}\sum_{i=1}^{n} \cos\ (2\pi x_i)\right) + 20 + e \tag{16}$$

Tables 1 and 2 summarize the benchmark functions and AFA parameters. Table 1 shows the functions used, their input ranges, and global optima. Table 2 lists the algorithm's settings for static and self-adaptive variants, including key constants and which parameters change adaptively, allowing a clear comparison between the two approaches.

**Table 1**
Functions used: Search space, initialization range, and optima

| **Function** | **Parameters** | | |
|---|---|---|---|
| | **Search Space** | **Initialization** | **Optima** |
| Sphere | -100≤ *xi* ≤100 | [-50,50] | $0.0^D$ |
| Rosenbrock | -30≤ *xi* ≤ 30 | [15,30] | $1.0^D$ |
| Schwefel 1.2 | -100≤ *xi* ≤100 | [50,100] | $0.0^D$ |
| Rastrigin | 5.12≤ *xi* ≤5.12 | [-2.56,2.56] | $0.0^D$ |
| Griewank | -600≤ *xi* ≤600 | [-300,600] | $0.0^D$ |
| Ackley | -32≤ *xi* ≤32 | [-16,32] | $0.0^D$ |

**Table 2**
Parameter settings for static and self-adaptive AFA variants

| Parameters | AFA Stage 1 Filter | AFA Stage 2 Filter | Self-Adaptive AFA Stage 1 Filter | Self-Adaptive AFA Stage 2 Filter |
|---|---|---|---|---|
| *k* cand (candidates per anchovy) | 5 | 5 | 5 | 5 |
| *m1* (top individuals for refinement) | - | 3 | - | 3 |
| *k* fine (refinement candidates per top individual) | - | 3 | - | 3 |
| S0 (initial step size) | 50 (fixed) | 50 (fixed) | lognormal init around 5 | lognormal init around 5 |
| *κ* (decay rate) | 3.0 | 3.0 | 3.0 | 3.0 |
| *ρ* (refinement divisor) | - | 5.0 | - | 5.0 |
| *γ* (exploration weight) | 0.5 (fixed) | 0.5 (fixed) | N(0.5,0.05), adaptive | N(0.5,0.05), adaptive |
| *λ* (local vs. global balance) | 0.7 (fixed) | 0.7 (fixed) | N(0.7,0.05), adaptive | N(0.7,0.05), adaptive |
| *β* (noise amplitude) | 0.01 (fixed) | 0.01 (fixed) | lognormal init around 0.01, adaptive | lognormal init around 0.01, adaptive |
| Adaptation mechanism | none | none | *γ λ,β,S0* evolve with probability τ =0.1 | γ,λ,β,S0 evolve with probability τ=0.1 |

## 3. Results

Experimental results obtained across six well-known benchmark functions such as Sphere, Rosenbrock, Schwefel 1.2, Rastrigin, Griewank and Ackley have provided meaningful insights into the comparative performance of the parameters of Static AFA (1-stage and 2-stage) and Self-Adapting AFA (1-stage and 2-stage). Convergence curves and statistical measures including mean, standard deviation, best and worst fitness values, as well as Wilcoxon signed-rank test results, highlight the strengths and weaknesses of the different algorithm variants. For the unimodal Sphere function, Self-Adapting AFA consistently outperforms Static AFA. Both 1-stage and 2-stage self-adaptation variants produce significantly lower mean errors with smaller deviations, indicating higher stability and reliability in convergence. The Wilcoxon test (p = 0.0019) confirms that the improvement of the self-adaptation approach is statistically significant.

**Table 3**
Simulation Results for Sphere Function

| Algorithm | Mean | Standard deviation | Best | Worst | p-value Vs static |
|---|---|---|---|---|---|
| Static AFA (1-stage) | 271.64 | 411.09 | 97.030 | 1491.5 | |
| Self-Adaptive AFA (1-stage) | 83.221 | 72.624 | 30.628 | 252.00 | 0.1054 |
| Static AFA (2-stage) | 119.94 | 50.947 | 90.313 | 271.26 | |
| Self-Adaptive AFA (2-stage) | 63.807 | 57.388 | 31.964 | 200.32 | 0.0839 |

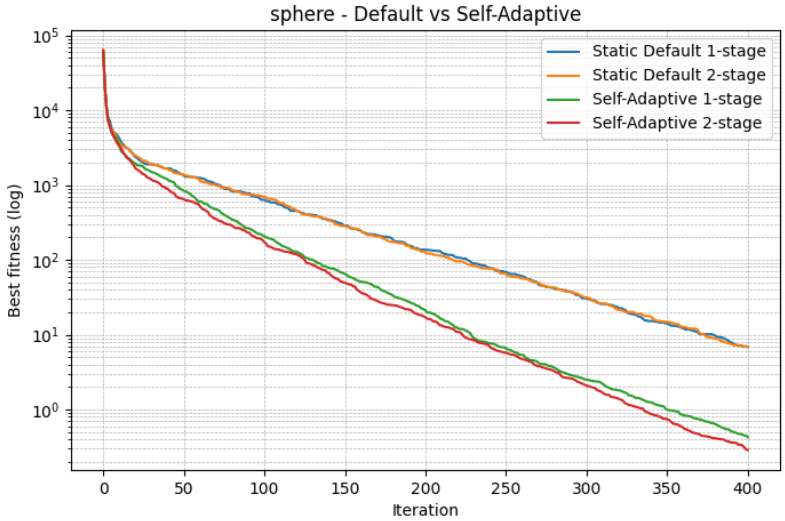


**Fig. 1.** Convergence curve comparison for Sphere Function

On the Rosenbrock function, which is characterized by a narrow-curved valley and strong variable dependency, the results show that Self-Adaptive AFA obtained lower mean fitness values than Static parameter AFA for both 1-stage and 2-stage variants. However, the standard deviation values suggest higher variability. The p-values (0.105 and 0.0839) indicate that improvements are not statistically significant, implying that while self-adaptive mechanisms improve exploration, their advantage over static variants in handling Rosenbrock's complexity is less conclusive.

**Table 4**
Simulation results for Rosenbrock Function

| Algorithm | Mean | Standard deviation | Best | Worst | p-value Vs static |
|---|---|---|---|---|---|
| Static AFA (1-stage) | 271.64 | 411.09 | 97.030 | 1491.5 | |
| Self-Adaptive AFA (1-stage) | 83.221 | 72.624 | 30.628 | 252.00 | 0.1054 |
| Static AFA (2-stage) | 119.94 | 50.947 | 90.313 | 271.26 | |
| Self-Adaptive AFA (2-stage) | 63.807 | 57.388 | 31.964 | 200.32 | 0.0839 |

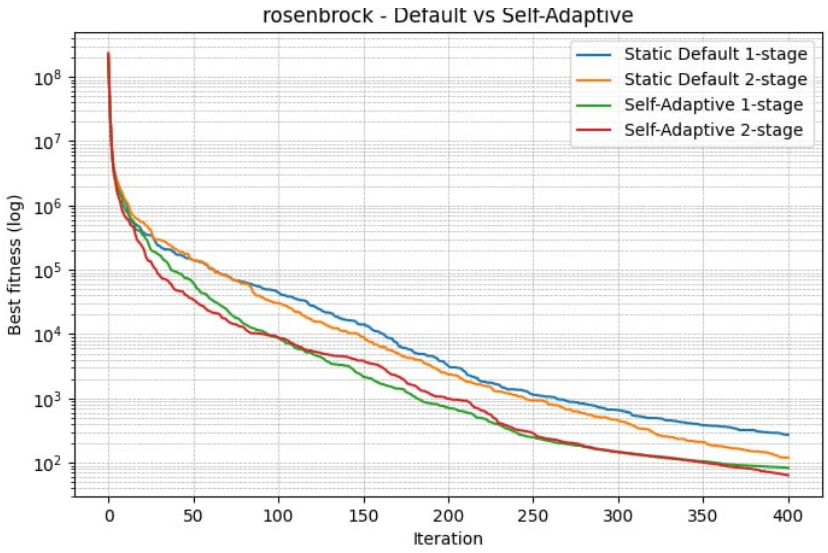


**Fig. 2.** Convergence curve comparison for Rosenbrock Function

In the Schwefel 1.2 test, which accumulates error across dimensions and penalizes deviations more severely, the Static AFA outperformed Self-Adaptive AFA in terms of mean and stability. Interestingly, the self-adaptive approach demonstrated higher best-case performance but was hindered by large deviations and poor worst-case results. This suggests that the adaptive mechanism may lead to premature convergence or instability in this function's landscape.

**Table 5**
Simulation results for Schewefel 1.2 Function

| Algorithm | Mean | Standard deviation | Best | Worst | p-value Vs static |
|---|---|---|---|---|---|
| Static AFA (1-stage) | 30.614 | 45.805 | 23.235 | 38.470 | |
| Self-Adaptive AFA (1-stage) | 55.666 | 27.216 | 30.066 | 12.672 | 0.0097 |
| Static AFA (2-stage) | 27.833 | 6.8865 | 15.848 | 40.095 | |
| Self-Adaptive AFA (2-stage) | 64.107 | 43.087 | 20.088 | 17.127 | 0.0097 |

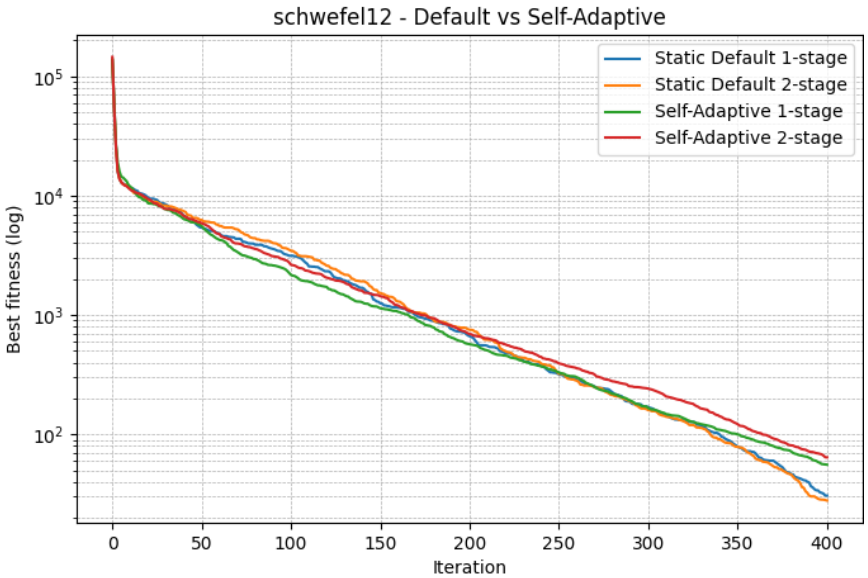


**Fig. 3.** Convergence curve comparison for Schewefel 1.2 Function

For the highly multimodal Rastrigin function, Self-Adaptive AFA (1-stage) achieved a better average result than Static AFA, while the 2-stage variants showed comparable outcomes. However, the p-values (0.1933 and 0.8457) indicate no significant statistical differences. These findings suggest that while self-adaptation aids in escaping local minima to some extent, the improvement over static approaches is modest in highly multimodal landscapes.

**Table 6**
Simulation results for Rastrigin Function

| Algorithm | Mean | Standard deviation | Best | Worst | p-value Vs static |
|---|---|---|---|---|---|
| Static AFA (1-stage) | 34.533 | 10.929 | 19.320 | 52.365 | |
| Self-Adaptive AFA (1-stage) | 26.299 | 6.8396 | 14.042 | 34.246 | 0.1933 |
| Static AFA (2-stage) | 28.838 | 6.3008 | 22.101 | 41.632 | |
| Self-Adaptive AFA (2-stage) | 28.177 | 7.6800 | 14.475 | 37.144 | 0.8457 |

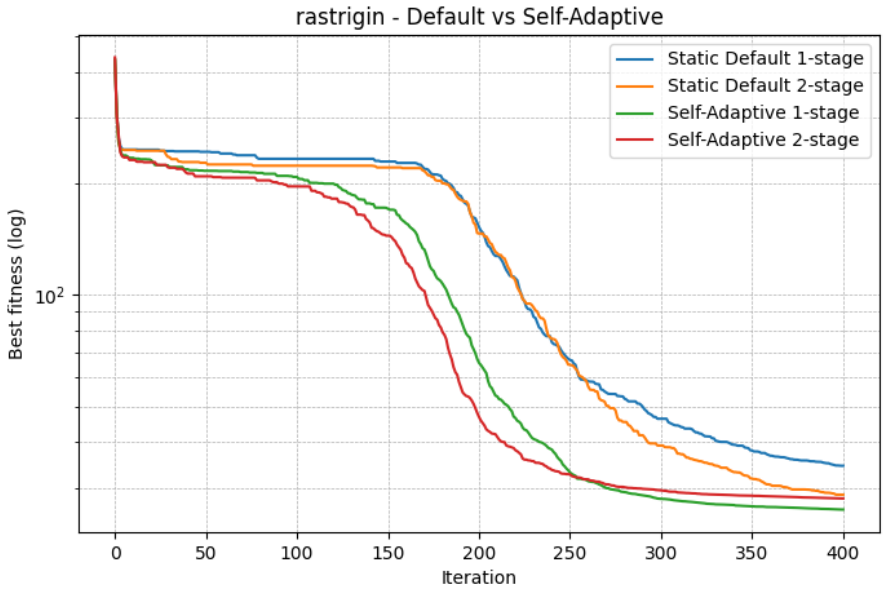


**Fig. 4.** Convergence curve comparison for Rastrigin Function

For the Griewank function, which is moderately multimodal with many widespread local minima, Self-Adaptive AFA clearly outperformed Static AFA. The reduction in mean fitness error was more than 50% for both 1-stage and 2-stage variants, with significantly lower standard deviation values. The Wilcoxon test confirmed statistical significance (p = 0.0019). This highlights the robustness of self-adaptive mechanisms in balancing exploration and exploitation across oscillatory search spaces.

**Table 7**
Simulation results for Griewank Function

| Algorithm | Mean | Standard deviation | Best | Worst | p-value Vs static |
|---|---|---|---|---|---|
| Static AFA (1-stage) | 1.0669, | 0.0083209 | 1.0590 | 1.0823 | |
| Self-Adaptive AFA (1-stage) | 0.55028 | 0.19969 | 0.20914 | 0.87176 | 0.0019 |
| Static AFA (2-stage) | 1.0645 | 0.0088033 | 1.0507 | 1.0834 | |
| Self-Adaptive AFA (2-stage) | 0.38779 | 0.096977 | 0.25018 | 53.699 | 0.0019 |

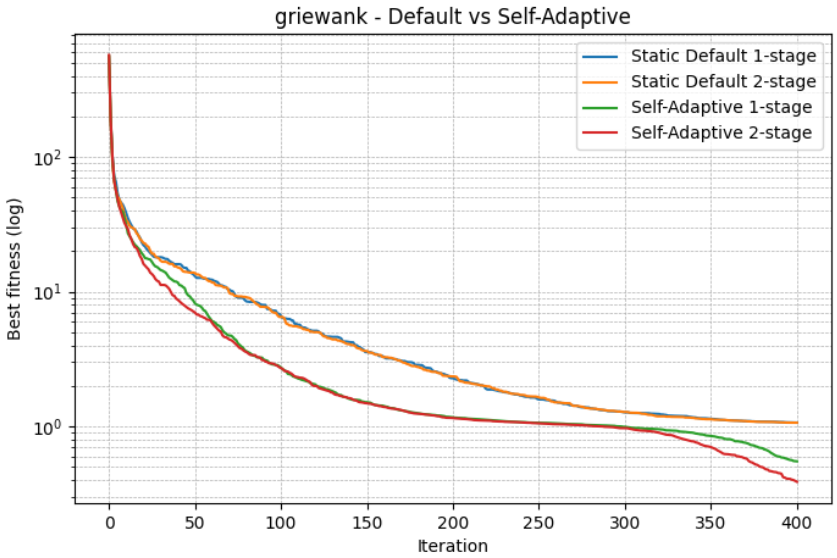


**Fig. 5.** Convergence curve comparison for Griewank Function

On the Ackley function, known for its large number of local minima surrounding the global minimum, Self-Adaptive AFA demonstrated a substantial improvement compared to Static AFA. Both 1-stage and 2-stage self-adaptive variants achieved dramatically lower mean and best fitness values, with the Wilcoxon test indicating high statistical significance (p = 0.0019). This confirms that self-adaptation effectively mitigates premature convergence and enables the algorithm to escape local minima more efficiently.

**Table 8**
Simulation results for Ackley Function

| Algorithm | Mean | Standard deviation | Best | Worst | p-value Vs static |
|---|---|---|---|---|---|
| Static AFA (1-stage) | 1.7070 | 0.17678 | 1.4438 | 2.0022 | |
| Self-Adaptive AFA (1-stage) | 0.26928 | 0.12028 | 0.097651 | 0.43222 | 0.0019 |
| Static AFA (2-stage) | 1.6785 | 0.15213 | 1.3674 | 1.8623 | |
| Self-Adaptive AFA (2-stage) | 0.22872 | 0.09.9396 | 0.1159 | 0.46064 | 0.001953 |

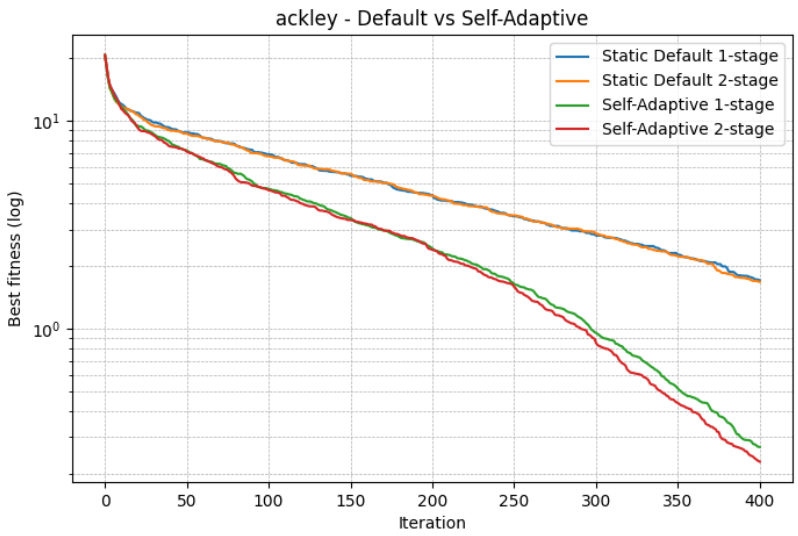


**Fig. 6.** Convergence curve comparison for Ackley Function

The comparative analysis reveals that Self-Adaptive AFA generally outperforms Static AFA, particularly for functions characterized by multimodality and rugged landscapes such as Griewank and Ackley. The improvements are statistically significant in these cases, proving the effectiveness of adaptive parameter control in enhancing convergence behavior. However, for unimodal or strongly correlated functions such as Rosenbrock and Schwefel 1.2, the advantage of self-adaptation is less consistent, and in some cases, Static AFA maintains superior stability.

These findings suggest that the strength of Self-Adaptive AFA lies in complex multimodal optimization problems, whereas Static AFA may remain a more reliable option for simpler unimodal landscapes. The convergence curve comparisons further confirm that self-adaptive variants not only converge faster but also maintain more consistent progress towards the global optimum in challenging problem spaces.

## 4. Conclusions

This paper introduced a Self-Adaptive Anchovy Filter Algorithm (AFA) that integrates parameter self-adaptation and multi-stage filtering mechanisms to enhance optimization performance. A comprehensive evaluation was conducted using six well-known benchmark functions, namely Sphere, Rosenbrock, Schwefel 1.2, Rastrigin, Griewank, and Ackley. The simulation results consistently demonstrated that the self-adaptive variants of AFA outperformed the static parameter AFA in terms of convergence speed and solution quality, particularly for unimodal functions such as Sphere and Ackley. The incorporation of dynamic parameter adaptation enabled a more effective balance between exploration and exploitation, as reflected in the significantly lower mean fitness values and Wilcoxon test outcomes ($p < 0.05$ in most cases). For multimodal and complex landscapes such as Rastrigin and Rosenbrock, the self-adaptive AFA achieved competitive results with improved robustness, though the statistical improvements were not always significant. This indicates that while self-adaptation enhances the search dynamics, further refinement may be required for functions with intricate local minima distributions. Overall, the proposed self-adaptive mechanism improved the stability, convergence behavior, and scalability of AFA across different benchmark functions. Future work will focus on hybridization with other evolutionary operators, application to real-world engineering problems, and further theoretical analysis of the convergence properties.

## Acknowledgement

Thanks to the Department of Electrical Engineering, Polytechnic Sultan Idris Shah for providing facilities and instruments during the experiment. We would also like to thank the laboratory assistant who helped us prepare the material in this study and not forgetting all those involved directly and indirectly in completing this study.

## References

[1] Mahfurdz, Azrul, Muhammad Muizz Mohd Nawawi, Sunardi Sunardi, and Mohd Azriq Abd Aziz. "Anchovy-inspired filter algorithm: A bio-inspired optimization approach for high-dimensional benchmark functions." *TELKOMNIKA (Telecommunication Computing Electronics and Control)* 24, no. 1 (2026): 271-281.https://doi.org/10.12928/telkomnika.v24i1.27594

[2] Holland, John H. 1975. *Adaptation in Natural and Artificial Systems*. University of Michigan Press. https://doi.org/10.7551/mitpress/1090.001.0001

[3] Goldberg, David E. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison Wesley.

[4] Kennedy, James, and Russell Eberhart. "Particle swarm optimization." In *Proceedings of ICNN'95-international conference on neural networks*, vol. 4, pp. 1942-1948. ieee, 1995. https://doi.org/10.1109/ICNN.1995.488968

[5] Shi, Yuhui, and Russell Eberhart. "A modified particle swarm optimizer." In *Evolutionary computation proceedings*, vol. 890, pp. 69-73. 1998. http://dx.doi.org/10.1109/icec.1998.699146
[6] Storn, Rainer, and Kenneth Price. "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces." *Journal of global optimization* 11, no. 4 (1997): 341-359. https://doi.org/10.1023/A:1008202821328
[7] Li, Guangqiang, Yawei Yang, Tinglu Zhao, Peixiang Peng, Yiran Zhou, Ying Hu, and Chen Guo. "An improved artificial fish swarm algorithm and its application to packing and layout problems." In *2017 36th Chinese Control Conference (CCC)*, pp. 9824-9828. IEEE, 2017. https://doi.org/10.23919/ChiCC.2017.8028923
[8] Bastos Filho, Carmelo JA, Fernando B. de Lima Neto, Anthony JCC Lins, Antonio IS Nascimento, and Marilia P. Lima. "A novel search algorithm based on fish school behavior." In *2008 IEEE international conference on systems, man and cybernetics*, pp. 2646-2651. IEEE, 2008. https://doi.org/10.1109/ICSMC.2008.4811695
[9] Blum, Christian, and Daniel Merkle. 2008. *Swarm Intelligence: Introduction and Applications*. Springer. https://doi.org/10.1007/978-3-540-74089-6
[10] Poli, Riccardo, James Kennedy, and Tim Blackwell. 2007. "Particle Swarm Optimization: An Overview." *Swarm Intelligence* 1 (1): 33–57. https://doi.org/10.1007/s11721-007-0002-0
[11] Bäck, Thomas. 1996. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press.
[12] Price, Kenneth V., Rainer M. Storn, and Jouni A. Lampinen. *Differential evolution: a practical approach to global optimization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. https://doi.org/10.1007/3-540-31306-0
[13] Derrac, Joaquín, Salvador García, Daniel Molina, and Francisco Herrera. "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms." *Swarm and evolutionary computation* 1, no. 1 (2011): 3-18. https://doi.org/10.1016/j.swevo.2011.02.002
[14] Eiben, Ágoston E., Robert Hinterding, and Zbigniew Michalewicz. "Parameter control in evolutionary algorithms." *IEEE Transactions on evolutionary computation* 3, no. 2 (1999): 124-141. https://doi.org/10.1109/4235.771166
[15] Lobo, F. J., Cláudio F. Lima, and Zbigniew Michalewicz, eds. *Parameter setting in evolutionary algorithms*. Vol. 54. Springer Science & Business Media, 2007. https://doi.org/10.1007/978-3-540-69432-8
[16] Karafotias, Giorgos, Mark Hoogendoorn, and Ágoston E. Eiben. "Parameter control in evolutionary algorithms: Trends and challenges." *IEEE Transactions on Evolutionary Computation* 19, no. 2 (2014): 167-187.https://doi.org/10.1109/TEVC.2014.2308294
[17] Aleti, Aldeida, and Irene Moser. "A systematic literature review of adaptive parameter control methods for evolutionary algorithms." *ACM Computing Surveys (CSUR)* 49, no. 3 (2016): 1-35. https://doi.org/10.1145/2996355
[18] Yang, Zhenyu, Ke Tang, and Xin Yao. "Self-adaptive differential evolution with neighborhood search." In *2008 IEEE congress on evolutionary computation (IEEE World Congress on Computational Intelligence)*, pp. 1110-1116. IEEE, 2008. https://doi.org/10.1109/CEC.2008.4630935
[19] Grefenstette, John J. "Optimization of control parameters for genetic algorithms." *IEEE Transactions on systems, man, and cybernetics* 16, no. 1 (1986): 122-128. https://doi.org/10.1109/TSMC.1986.289288
[20] Beyer, Hans-Georg, and Hans-Paul Schwefel. "Evolution strategies–a comprehensive introduction." *Natural computing* 1, no. 1 (2002): 3-52. https://doi.org/10.1023/A:1015059928466
[21] Brest, Janez, Sao Greiner, Borko Boskovic, Marjan Mernik, and Viljem Zumer. "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems." *IEEE transactions on evolutionary computation* 10, no. 6 (2006): 646-657. https://doi.org/10.1109/TEVC.2006.872133
[22] Zhang, Jingqiao, and Arthur C. Sanderson. "JADE: adaptive differential evolution with optional external archive." *IEEE Transactions on evolutionary computation* 13, no. 5 (2009): 945-958. https://doi.org/10.1109/TEVC.2009.2014613
[23] Tanabe, Ryoji, and Alex Fukunaga. "Success-history based parameter adaptation for differential evolution." In *2013 IEEE congress on evolutionary computation*, pp. 71-78. IEEE, 2013.https://doi.org/10.1109/CEC.2013.6557555
[24] Mallipeddi, Rammohan, Ponnuthurai N. Suganthan, Quan-Ke Pan, and Mehmet Fatih Tasgetiren. "Differential evolution algorithm with ensemble of parameters and mutation strategies." *Applied soft computing* 11, no. 2 (2011): 1679-1696. https://doi.org/10.1007/978-3-642-17563-3_9
[25] Hansen, Nikolaus, and Andreas Ostermeier. "Completely derandomized self-adaptation in evolution strategies." *Evolutionary computation* 9, no. 2 (2001): 159-195. https://doi.org/10.1162/106365601750190398
[26] Mendes, Rui, James Kennedy, and José Neves. "The fully informed particle swarm: simpler, maybe better." *IEEE transactions on evolutionary computation* 8, no. 3 (2004): 204-210.https://doi.org/10.1109/TEVC.2004.826074

[27] Bastos-Filho, C. J. A., and D. O. Nascimento. "An enhanced fish school search algorithm." In *2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence*, pp. 152-157. IEEE, 2013. https://doi.org/10.1109/BRICS-CCI-CBIC.2013.34

[28] Omran, Mahamed GH, and Mehrdad Mahdavi. "Global-best harmony search." *Applied mathematics and computation* 198, no. 2 (2008): 643-656. https://doi.org/10.1016/j.amc.2007.09.004

[29] Das, Swagatam, and Ponnuthurai Nagaratnam Suganthan. "Differential evolution: A survey of the state-of-the-art." *IEEE transactions on evolutionary computation* 15, no. 1 (2010): 4-31. https://doi.org/10.1109/TEVC.2010.2059031

[30] Gao, Mingqiang, and Xu Yang. "APSO-SL: An adaptive particle swarm optimization with state-based learning strategy." *Processes* 12, no. 2 (2024): 400. https://doi.org/10.3390/pr12020400