OKÜ Fen Bilimleri Enstitüsü Dergisi
8(1): 325-352, 2025

OKU Journal of The Institute of Science and Technology, 8(1): 325-352, 2025

Osmaniye Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü Dergisi

Osmaniye Korkut Ata University Journal of The Institute of Science and Technology

Afrika Akbabaları Optimizasyon Algoritmasının Güncel Metasezgisellerle Karşılaştırmalı Analizi

Sibel ARSLAN[1*], Yıldız ZORALİOĞLU[2], Muhammed Furkan GÜL[3]

[1]Sivas Cumhuriyet University, Faculty of Technology, Department of Software Engineering, Sivas
[2]Sivas Cumhuriyet University, Faculty of Engineering, Department of Computer Engineering, Sivas
[3]Sivas University of Science and Technology, Faculty of Engineering and Natural Sciences, Department of Computer Engineering, Sivas

[1]https://orcid.org/0000-0003-3626-553X
[2]https://orcid.org/0009-0008-7482-0964
[3]https://orcid.org/0009-0007-0486-0525
*Sorumlu yazar: sibelarslan@cumhuriyet.edu.tr

Araştırma Makalesi

Makale Tarihçesi:
Geliş tarihi: 13.05.2024
Kabul tarihi: 13.09.2024
Online Yayınlanma: 15.01.2025

Anahtar Kelimeler:
Optimizasyon
Metasezgisel algoritma
Test fonksiyonları

ÖZ

Optimizasyon problemlerinin karmaşıklığının artmasıyla birlikte yeni metasezgisel algoritmalar geliştirilmektedir. Bu algoritmalar farklı problemler üzerinde üstün performanslar sergileyerek başarılarını göstermektedir. Bu çalışmada; son zamanlarda önerilen dört metasezgisel algoritma olan Yapay Sinekkuşu Algoritması (Artificial Hummingbird Algorithm, AHA), Afrika Akbabaları Optimizasyon Algoritması (African Vultures Optimization Algorithm, AVOA), Kerevit Optimizasyon Algoritması (Crayfish Optimization Algorithm, COA) ve Deniz Yırtıcıları Optimizasyon Algoritması'nın (Marine Predators Optimization Algorithm, MPA) 26 test fonksiyonu üzerindeki performansları karşılaştırılmıştır. Karşılaştırmalar sonucunda algoritmaların farklı fonksiyonlar üzerinde çok küçük farklarla birbirlerinden daha iyi performans gösterdiği gözlemlenmiştir. Aynı zamanda karşılaştırma sonuçları t-test istatistiksel testi ile değerlendirilmiştir. AVOA, çeşitli test fonksiyonları için çözümlerin kalitesini değerlendirmede diğer yeni metasezgisellere göre daha iyi veya karşılaştırılabilir performans göstermiştir. Gelecek araştırmalarda AVOA'nın farklı problemler üzerinde kullanılması hedeflenmektedir.

A Comparative Analysis of African Vultures Optimization Algorithm with Current Metaheuristics

Research Article

Article History:
Received: 13.05.2024
Accepted: 13.09.2024
Published online: 15.01.2025

Keywords:
Optimization
Metaheuristic algorithm
Test functions

ABSTRACT

With the increasing complexity of optimization problems, new metaheuristic algorithms are being developed. These algorithms show their success by exhibiting superior performances on different problems. In this paper, the performance of four recently proposed metaheuristic algorithms, namely Artificial Hummingbird Algorithm (AHA), African Vultures Optimization Algorithm (AVOA), Crayfish Optimization Algorithm (COA) and Marine Predators Optimization Algorithm (MPA) on 26 test functions are compared. As a result of the comparisons, it was observed that the algorithms outperformed each other with very small differences on different functions. At the same time, the comparison results were evaluated by t-test statistical test. AVOA has shown better or comparable performance to other recent metaheuristics in evaluating the quality of solutions for several test functions. It is aimed to use AVOA on different problems in future research.

To Cite: Arslan S., Zoralioğlu Y., Gül MF. A Comparative Analysis of African Vultures Optimization Algorithm with Current Metaheuristics. Osmaniye Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü Dergisi 2025; 8(1): 325-352.

1. Introduction

Over the past few years, the difficulty of optimization problems has increased with the advancement of modern technology. Moreover, many metaheuristic algorithms inspired by nature have been developed to solve many optimization problems (Zhoa et al., 2022; Azizi et al., 2023; Deng et al., 2023; Ghaedi et al., 2023; Zhu et al., 2023). These algorithms solve large-scale problems by returning near-optimal results. In addition, newly developed metaheuristic algorithms prove their success by providing great superiority over the popular ones (Arslan, 2023). Among the popular metaheuristic algorithms, Genetic Algorithm (GA) is based on the principle of survival of the fittest (Mirjalili, 2019). Another famous algorithm in swarm-based population is Particle Swarm Optimization (PSO) (Kennedy et al. 1995). Artificial Bee Colony (ABC) (Karaboga, 2010), Ant Colony Optimization (ACO) (Dorigo et al., 2006) and Simulated Annealing (SA) (Dowsland et al., 2012) are other popular population-based algorithms. Evaluating the performance of both these popular algorithms and the newly developed metaheuristics among themselves allows us to find the most suitable algorithm for solving the problems. For this purpose, it is necessary to analyze the performance of newly developed metaheuristics. One of the goals of this paper is to observe the effectiveness of the newly developed Artificial Hummingbird Algorithm (AHA) (Zhao et al., 2022), African Vultures Optimization Algorithm (AVOA) (Abdollahadeh et al., 2021), Crayfish Optimization Algorithm (COA) (Jia et al., 2023) and Marine Predator Optimization Algorithm (MPA) (Faramarzi et al., 2020). Another goal of this paper is to determine the most successful algorithm among the algorithms tested on 26 test functions to solve real-world problems. Other contributions of the paper are as follows:

- To the best of our knowledge, there is no comparison of these four new metaheuristic algorithms in the literature.
- 26 test functions were used to analyze their success.
- According to the information obtained from the experimental results, AVOA showed the best success.

In the rest of the paper, Section 2 gives detailed information about the algorithms and describes their pseudocode or flowcharts. Section 3 presents the test functions used and the parameters of the algorithms. Section 4 provides information about the experimental results, convergence graphs and t-test results. Section 5 provides conclusions and future work.

2. Algorithms

In this section, newly proposed metaheuristics are explained in detail.

2.1. Artificial Hummingbird Algorithm (AHA)

AHA is developed by Zhao et al. in 2022 inspired by the behavior of the hummingbird (Zhao et al., 2022). The metaheuristic modelled the flight skills, memory capacity and foraging strategies of hummingbirds. First, a hummingbird evaluates the characteristics of its food sources, including the content of individual flowers, nectar quality, nectar refilling and the time of last visit. In AHA, each food source is assumed to have the same number and type of flowers. A food source is a solution vector and the nectar filling rate of the source indicates the fitness value of the function. A good fitness value is associated with a high nectar filling rate of the food source. Since each hummingbird is assigned to a specific food source, the hummingbird and the food source have the same position. The hummingbird can memorize the position and nectar filling rate of this particular food source and share it with other hummingbirds. The level of frequentation of each food source by different hummingbirds is recorded in a visit table. If this level of visit is high, that food is prioritized. Likewise, to obtain more nectar, the food with the highest nectar filling rate among the food sources with the same highest visitation level is preferred. A visit table is used to find the target food source and this table is updated at each iteration. Based on the behaviors described, there are guided foraging, territorial foraging and migration foraging phases.

2.1.1. Guided foraging

The tendency to visit the food source with the maximum nectar volume means that it has a high nectar filling rate and is not visited much. At this phase, the food sources with the highest level of visit are expected to be identified and then the one with the highest nectar filling rate is selected as the target food source. Once the target food source has been selected, the hummingbird can fly towards it to feed. Three flight skills are used and modelled during foraging in the AHA: omnidirectional, directional and axial flights. These flight patterns can be extended to a d -D domain. The mathematical model of axial flight is as given in Eq 1.

$$D^{(i)} = \begin{cases} 1 & \text{if } i = \text{randi}\,([1, d]) \\ 0 & \text{else} \end{cases} \quad i = 1, \dots, d \tag{1}$$

The models for directional and omnidirectional flights are defined as in Eq 2-3 respectively.

$$D^{(i)} = \begin{cases} 1 & \text{if} \\ 0 & \text{else} \end{cases} \quad i = P(j), j \in [1, k], P = randperm(k), k \in [2, \lceil r_1 \cdot (d-2) \rceil + 1] \; i = 1, \dots, d \tag{2}$$

$$D^{(i)} = 1 \; i = 1, \dots, d \tag{3}$$

where $\text{randi}\,([1, d])$ generates a random number from the number range from 1 to d. $randperm(k)$ produces a random integer permutation from the numbers from 1 to k. r_1 a randomly generated number between [0,1]. The mathematical model representing the guided foraging behaviour and candidate food source is given in Eq. 4-5.

$$v_i(t+1) = x_{i,\text{tar}}(t) + a \cdot D \cdot \left(x_i(t) - x_{i,\text{tar}}(t)\right) \tag{4}$$

$$a \sim N(0,1) \tag{5}$$

where $x_i(t)$ represents the position of the ith food source at time t. $x_{i,\text{tar}}(t)$ is the position of the target food source, and the ith bird is planning to visit this source. a is the guided factor and follows a normal distribution. The update of the position of the ith food source is modeled in Eq. 6.

$$x_i(t+1) = \begin{cases} x_i(t) & f(x_i(t)) \leq f(v_i(t+1)) \\ v_i(t+1) & f(x_i(t)) > f(v_i(t+1)) \end{cases} \tag{6}$$

where $f(.)$ is the value of the fitness function. In this model, if the nectar filling rate of the candidate food source is better compared to the previous one, it indicates that the hummingbird abandons the previous food source and stays with the candidate food source in Eq. 4. According to the visit table, a hummingbird can find its desired target food source in each iteration. The visit table keeps track of how long each food source has been unvisited by the same hummingbird since the last time it was visited. A long absence indicates a high visit level. Therefore, each hummingbird wants to find the food source according to the high visit level. If more than one source has the same level, the one with the best nectar filling rate is selected. Each hummingbird in the population travels to the target food source using Eq. 4 to the target food source. When guided foraging occurs in each iteration, the visit levels for that hummingbird are increased by one. At the same time, the visit level of the visited target food source is initialized to zero. Once this step has been performed, the food source is not changed, unless a better nectar filling rate solution is available. Updating a food source where the hummingbird of interest resides means updating the visit level for the others. The visit level to be updated is expressed as the highest level of the other food sources increased by 1.

2.1.2. Territorial foraging

After visiting the target food source where the flower nectar is consumed, a new food source can be sought to visit other available food sources. Therefore, the hummingbird can easily move to a neighbouring region within its territory, where it can find a new food source as a candidate solution that may be better than the current solution. The mathematical model of the territorial search and candidate food source at this stage is given in Eq. 7-8.

$$v_i(t+1) = x_i(t) + \mathrm{b} \cdot D \cdot (x_i(t)) \tag{7}$$

$$\mathrm{b} \sim N(0,1) \tag{8}$$

where b is the territorial factor and has a normal distribution. After this step, the visit table should be updated.

2.1.3. Migration foraging

In the event of food shortages in the area that the hummingbird usually visits, it migrates to another distant food source. A migration coefficient is defined in the AHA. If the number of iterations exceeds the predefined value of the migration coefficient, the hummingbird that settled on the food source exhibiting the lowest rate of nectar filling will migrate to a new food source. After the hummingbird leaves the food source, the visit table will be updated. The modelling of foraging from the source with the worst nectar filling rate towards the new randomly generated source is given in Eq. 9.

$$x_{wor}(t+1) = Low + r \cdot (Up - Low) \tag{9}$$

where x_{wor} is the ratio with the worst nectar filling in the population. Low and Up are the lower and upper bounds. The flowchart of AHA is given in Figure 1.

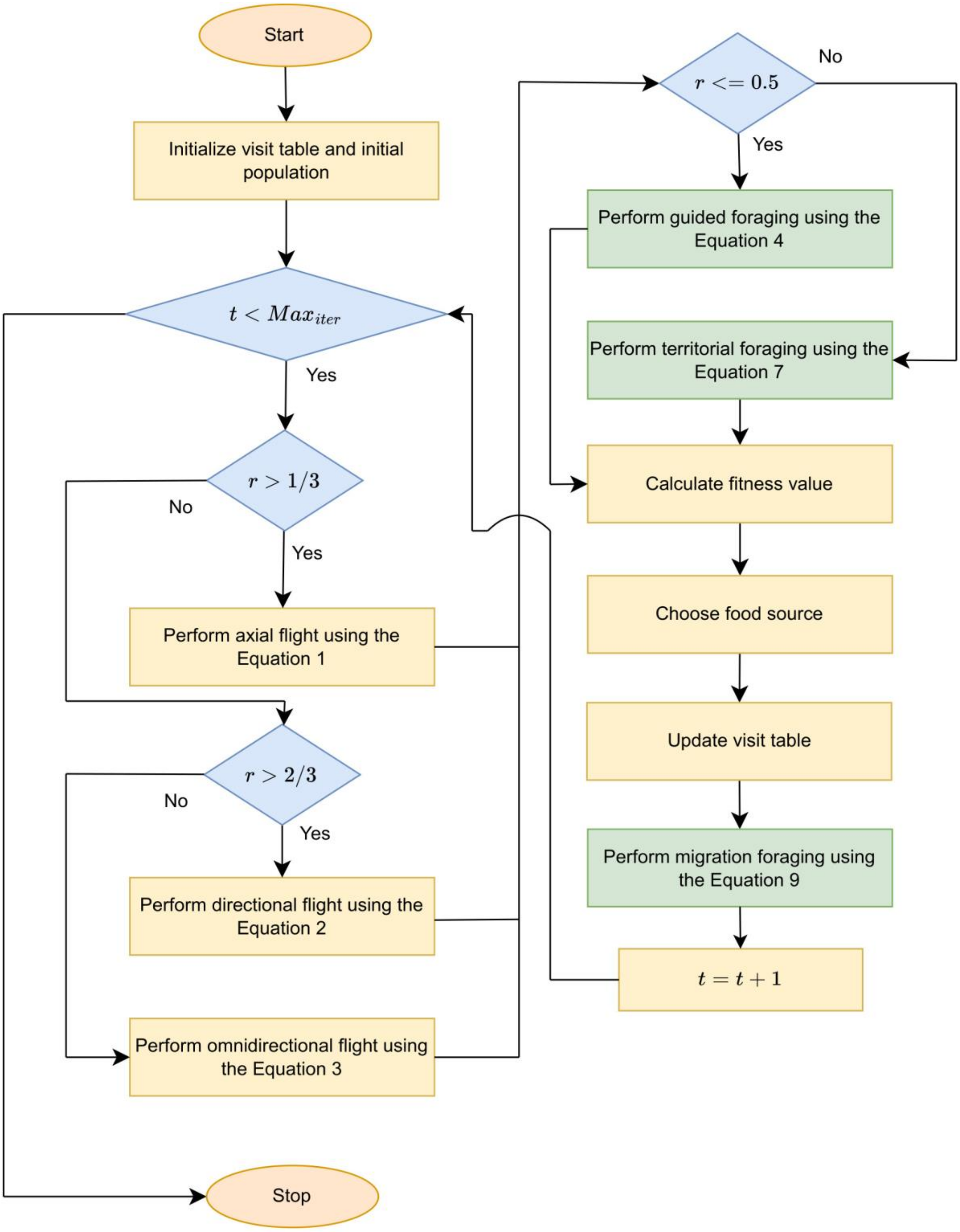

Figure 1. The flowchart of AHA (Hosseinzadeh et al., 2024)

2.2. African Vultures Optimization Algorithm (AVOA)

AVOA is a recent metaheuristic developed by Abdollahzadeh et al (Abdollahzadeh et al., 2021). Based on the foraging and location behavior of vultures. In this metaheuristic, each vulture represents a solution, and the fitness value is calculated from all solutions. According to this fitness value, vultures are categorized into two basic groups to find food and survive as a group. In the first group, the vulture with the highest value is considered the first and best vulture of the group, and in the second group, the vulture with the second highest value is considered the first and best vulture of the group. The other vultures are used to move or replace these two best vultures. In AVOA, the worst solution is considered to be the weakest and hungry vulture, and they try to avoid this vulture and get closer to the best vulture. AVOA consists of four phases: identifying the best vulture in the group, calculating the hunger rate of vultures, search and exploitation.

2.2.1. Identifying the best vulture in the group

The initial population is generated, and the fitness values of all solutions are computed. The top-performing solution in the first group is considered the best, while the second-best solution is identified in the second group. The remaining solutions in both groups adjust their positions towards these top solutions. In each iteration, the shift is determined based on the computed fitness values.

2.2.2. Calculating the hunger rate of vultures

Vultures have the most energy when they are full and can fly long distances in search of food (Xue et al., 2018). However, when they are not full, they cannot fly for a long time in search of food because they do not have enough energy. To improve performance in solving complex problems, Eq. 10 is modelled.

$$t = h \times \left(sin^{w} \left(\frac{\pi}{2} \times \frac{\text{iter}}{max_{iter}} \right) + cos \left(\frac{\pi}{2} \times \frac{\text{iter}}{max_{iter}} \right) - 1 \right) \tag{10}$$

where h is obtained from random values between -2 and 2. $iter$ and max_{iter} refer to the current and the maximum iteration, respectively. Hunger rates of vultures, which tend to decrease, are modelled in Eq. 11.

$$F = (2rand_1 + 1) \times z \times \left(1 - \frac{\text{iter}}{max_{iter}} \right) + t \tag{11}$$

where z represents a randomly chosen number within the range of -1 to 1 and changes at each iteration. $rand_1$ is a parameter that takes random values between 0 and 1. If z is less than 0, it indicates the vulture is hungry, whereas if it exceeds 0, it signifies the vulture is full. If the F value is greater than or equal to 1, vultures search for food in various locations to feed and the exploration phase in AVOA begins. If this value is less than 1, vultures forage in neighbouring solutions and the exploitation phase begins.

2.2.3. Exploration

In this phase, the vultures choose two strategies and search different areas using the parameter p_1. This parameter is evaluated before the search process and takes a value between 0 and 1. For the selection of strategies, Eq. 12 is used.

$$P_i(t+1) = \begin{cases} R(i) - |X \times R(i) - P(i)| \times F \\ \quad if\ p_1 \geq rand_{P_1} \\ R(i) - F + rand_2 \times \left((u_b - l_b) \times rand_3 + l_b\right) \\ \quad if\ p_1 < rand_{P_1} \end{cases} \tag{12}$$

where $R(i)$ represents one of the best vultures. The parameter X represents the distance it travelled to protect the food, $rand_2$ and $rand_3$ are random numbers between [0,1]. The current vector position, represented by $P(i)$, indicates the distance. The lower boundary of the search space is l_b and the upper boundary is u_b. As the value of $rand_3$ gets closer to 1, the search for different space areas and diversity in AVOA increases. The parameter $rand_{P_1}$ takes a random value between [0,1] to select one of the two strategies.

2.2.4. Exploitation

When the parameter F, i.e. the hunger rate, is less than 1, AVOA enters the exploitation. This phase is separated into two selections according to whether F is smaller or larger than 0.5. These two choices depend on the values of the randomly generated p_2 and p_3. In the first choice, the vultures are full and have enough energy. This may lead them into conflicts over food acquisition. In the second selection, the behavior of the vultures makes the various vulture species aggressive in besieging and foraging on the food source. These behaviors depend on the parameter p_2 and are modelled in Eq. 13.

$$P_i(t+1) = \begin{cases} |X \times R(i) - P(i)| \times \\ \quad if\ p_2 \geq rand_{P_2} \\ R(i) - R(i) \times \left(\dfrac{P(i)}{2\pi}\right)\begin{pmatrix} rand_5 \\ \times \cos\big(P(i)\big) \\ + rand_6 \times \sin\big(P(i)\big) \end{pmatrix} \\ \quad if\ p_2 < rand_{P_2} \end{cases} \tag{13}$$

In Eq. 13, parameters $rand_4$, $rand_5$ and $rand_6$ take randomly generated values between 0 and 1. If the parameter F is less than 0.5, the second selection of this phase takes place. In this selection, a random value between 0 and 1 is generated for $rand_{P_3}$. If the parameter p_3 is greater than or equal to $rand_{P_3}$, multiple vultures gather on the food source and competition occurs between them to get the food. In the other case, a siege war is fought between the vultures. The mathematical model of this selection is shown in Eq. 14.

$$P(i+1) = \begin{cases} Eq.16\ if\ \ p_3 \geq rand_{P_3} \\ Eq.17\ if\ \ p_3 < rand_{P_3} \end{cases} \tag{14}$$

The movement of the vultures to congregate on the food source they have found is described in Eq. 15 is expressed in Eq. 16.

$$
\begin{aligned}
A_1 = & \text{ BestVulture }_1(i) \\
& -\frac{\text{BestVulture }_1(i) \times P(i)}{\text{BestVulture }_1(i)-P(i)^2} \times F \\
A_2 = & \text{ BestVulture }_2(i) \\
& -\frac{\text{BestVulture }_2(i) \times P(i)}{\text{BestVulture }_2(i)-P(i)^2} \times F
\end{aligned} \tag{15}
$$

where, in the current iteration, $\text{BestVulture }_1(i)$ and $\text{BestVulture }_2(i)$ are the best vultures.

$$P(i+1)=\frac{A_1+A_2}{2} \tag{16}$$

When $F < 0.5$, the leader vultures of the groups are hungry and do not have enough energy to cope with the group members. In this case, the leader vultures move in different directions. Eq. 17 gives a model of the movement.

$$P(i+1)=R(i)-|R(i)-P(i)| \times F \times \text{LF }(d) \tag{17}$$

where d is the problem size, $|R(i)-P(i)|$ is the distance of the vulture. To increase the efficiency of AVOA, Levy Flight (LF) (Yang, 2010; Yang, 2009) was also included. Modelling of this flight is as in Eq. 18.

$$LF(x)=0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}}, \sigma\left(\frac{\Gamma(1+\beta) \times \sin \left(\frac{\pi \rho}{2}\right)}{\Gamma(1+\beta 2) \times \beta \times 2\left(\frac{\beta-1}{2}\right)}\right)^{\frac{1}{\beta}} \tag{18}$$

where u and v are random numbers generated between 0 and 1 and β is a constant value set at a default of 1.5. The flowchart of AVOA is given in Figure 2.

2.3. Crayfish Optimization Algorithm (COA)

COA is a proposed by Jia et al. in 2023, inspired by the behavior of crayfish (Jia et al., 2023). The metaheuristic was developed in three stages, based on the behaviors of summer holiday destination, competition and foraging. In the holiday destination phase, the crayfish find a cave for a holiday and settle down. When there is more than one crayfish, there is a competition for the found cave. This behavior is modelled as the competition phase. Foraging behavior is the stage in which the food is broken down according to the size of the food. The transition between these stages depends on the air temperature. The optimum air temperature range is 20-30°C and the amount of feeding of crayfish can be affected by temperature. The temperature of the crayfish's environment is defined in Eq. 19.

$$temp = rand \times 15 + 20 \tag{19}$$

where $temp$ is the temperature of the environment in which the crayfish are found. The mathematical model of crayfish intake as a function of temperature is given in Eq. 20.

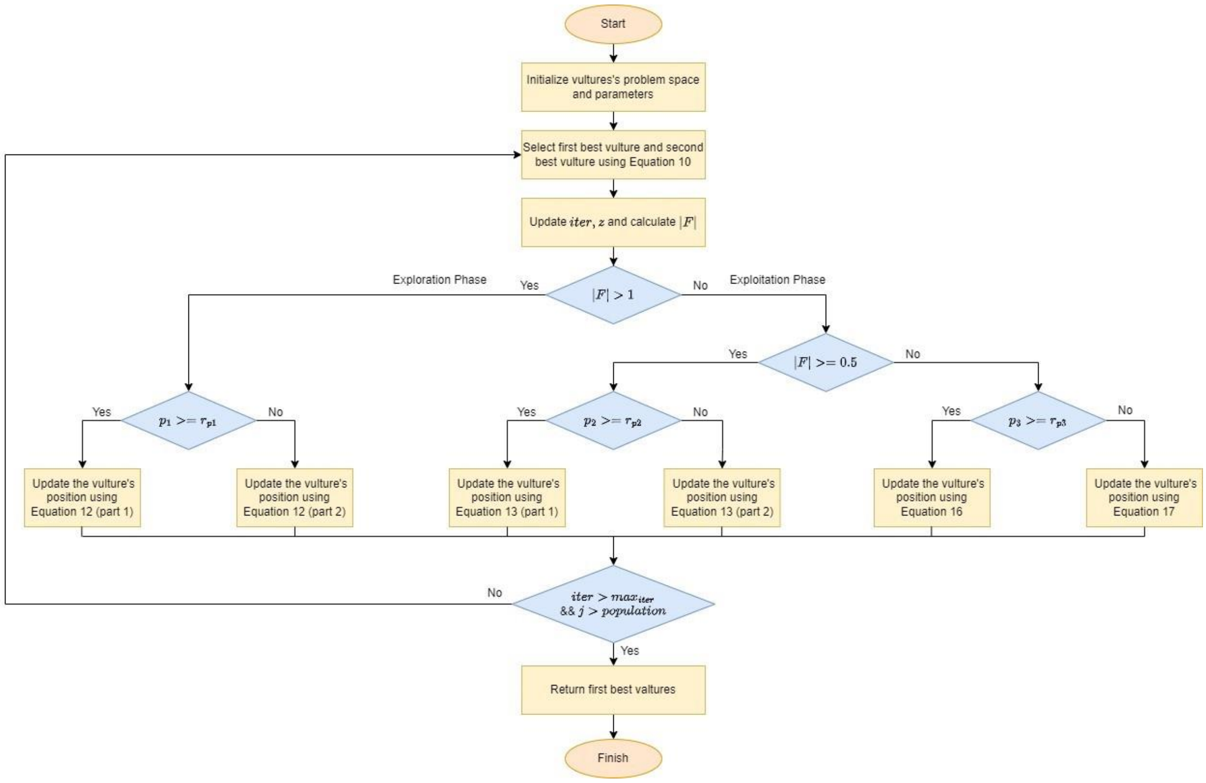

Figure 2. The flowchart of AVOA (Abdollahadeh et al., 2021)

$$p = C_1 \times \left(\frac{1}{\sqrt{2 \times \pi} \times \sigma)} \times \exp\left(-\frac{(temp - \mu)^2}{2\sigma^2} \right) \right) \tag{20}$$

where σ and C_1 control the uptake of crayfish at different temperatures, while μ is the optimum temperature for crayfish. The stages of COA are described in the following sections.

2.3.1. Summer holiday place phases (Exploration)

When the temperature is above 30°C, crayfish go on holiday to explore caves (Jia et al., 2023). This expands the solution space of the algorithm. The mathematical equation of the cave, represented by X_{shade} , is determined as provided in Eq. 21.

$$X_{\text{shade}} = (X_G + X_L)/2 \tag{21}$$

where X_G represents the best position so far with respect to the number of iterations, while X_L represents the best position of the current population. The parameter X_{shade} used in Eq. 3 is taken to model Eq. 22 for entering the cave.

$$X_{i,j}^{t+1} = X_{i,j}^t + C_2 \times \text{rand} \times \left(X_{\text{shade}} - X_{i,j}^t\right) \tag{22}$$

where, $X_{i,j}$ represents the position of the ith individual in the jth dimension. t denotes the current iteration, and $t + 1$ represents the next iteration. C_2 is the decay curve defined in Eq. 23.

$$C_2 = 2 - (t/T) \tag{23}$$

where T is the maximum iteration. When the random rand is less than 0.5, it means that there are no other crayfish competing for the cave. In this way, the crayfish enters the cave directly without competing for a holiday. In COA, when the random number is equal to or exceeds 0.5, the competition phase initiates.

2.3.2. Competition phase (Exploitation)

Exploitation is choosing the best among the existing solutions. A crayfish enters a competitive phase and performs exploitation in order not to give the existing cave it has found to another crayfish. When there is more than one crayfish interested in the cave, the temperature is more than 30°C and the random number is greater than or equal to 0.5. In this case, the crayfish compete for the cave. The mathematical model of this competition is given in Eq. 24.

$$X_{i,j}^{t+1} = X_{i,j}^{t} - X_{z,j}^{t} + X_{\mathrm{shade}} \tag{24}$$

Crayfish in X_i update their position relative to another crayfish in X_z. This improves the exploration capability and extends the range of the COA. A random crayfish is represented by z.

2.3.3. Foraging Phase (Exploitation)

In the foraging phase COA develops the exploitation by approaching the optimum solution. In this exploitation, the temperature must be less than or equal to 30°C for the crayfish to feed. When they find food, they decide according to its size. If the food is small, they eat it directly, but if it is large, they shred it up and eat it with their second and third legs. The food size, Q, is given in Eq. 25.

$$Q = C_3 \times rand \times (fitness_i / \, fitness_{food}) \tag{25}$$

The constant C_3 represents the largest food and its defined value for the algorithm is 3. $fitness_i$ denotes the fitness value of the ith crayfish, while $fitness_{food}$ represents the fitness value of the food position. It is determined that the food is very large if $Q > (C_3 + 1)/2$. In this case, the crayfish will shred the food with its first leg. The mathematical model for this fragmentation is given in Eq. 26.

$$X_{\mathrm{food}} = \exp\left(-\frac{1}{\mathrm{Q}}\right) \times X_{\mathrm{food}} \tag{26}$$

where X_{food} represents the optimal solution. When the food is cut into small pieces, it is taken into the mouth using the second and third legs. The search equation for the food eaten by the crayfish is given by Eq. 27. When $Q \leq (C_3 + 1)/2$, the crayfish moves towards the food and eats it with its first leg. This situation is modelled in Eq. 28. COA's pseudo code is given in Algorithm 1.

$$X_{i,j}^{t+1} = X_{i,j}^{t} + X_{\mathrm{food}} \times p \times (\cos(2 \times \pi \times rand) - \sin(2 \times \pi \times rand)) \tag{27}$$

$$X_{i,j}^{t+1} = \left(X_{i,j}^{t} - X_{\mathrm{food}}\right) \times p + p \times \mathrm{rand} \times X_{i,j}^{t} \tag{28}$$

Algorithm 1 COA

1. Initialize parameters of COA
2. Randomly generate an initial population
3. Calculate the fitness value of the population to get $X_G,\ X_L$
4. **while** ($iter\ < max_{iter}$)
5. Defining temp by Eq. 19
6. **if** $(temp >\ 30)$
7. Define cave X_{shade} according to Eq. 21
8. **if** $(rand\ <\ 0.5\ )$
9. Crayfish conducts the summer resort stage according to Eq. 22
10. **else**
11. Crayfish compote for caves through Eq. 24
12. **end if**
13. **else**
14. The food size Q and food intake p are obtained by Eq. 20 and Eq. 25
15. **if** $(Q\ >\ 2\ )$
16. Crayfish shreds food by Eq. 26
17. Crayfish foraging according to Eq. 27
18. **else**
19. Crayfish foraging according to Eq. 28
20. **end if**
21. **end if**
22. Update fitness values, $X_G,\ X_L$
23. $iter$ = $iter$ + 1
24. **end while**

2.4. Marine Predators Algorithm (MPA)

MPA is a new metaheuristic developed by Faramarzi et al. in 2020 inspired by the attack behavior of marine predators on their prey (Faramarzi et al., 2020). In the metaheuristic, predators use two basic motions, Brownian and Levy, when targeting their prey. In order to achieve a balance between these two motions, sea predators calculate a velocity ratio based on their current position relative to the prey. MPA initially generates solutions by randomly distributing them within the search space as defined in Eq. 29.

$$X_0 = X_{min} + B(X_{max} - X_{min}) \tag{29}$$

where X_{min} and X_{max} are the minimum and maximum limits for the search space. The parameter B takes randomly generated values between 0 and 1. MPA consists of Elite and Prey matrices. The Elite matrix identifies the best hunter. Thus, the hunter ensures that the prey is searched and detected by using its location. This matrix is modelled in Eq. 30.

$$\text{Elite} = \begin{bmatrix} X_{1,1}^I & \cdots & X_{1,d}^I \\ \vdots & \ddots & \vdots \\ X_{n,1}^I & \cdots & X_{n,d}^I \end{bmatrix} \tag{30}$$

where n and d are the population and problem size respectively. $\overrightarrow{X^I}$ is defined as the best sea predator that forms the Elite matrix with n iterations. This matrix is updated as the predator may change because of each iteration.

In the MPA, the population consists of predators and prey. The main reason for this formation is that while the predator is looking for prey, it can also become prey for other predators. To update the positions of the predators, the Prey matrix given in Eq. 31 is used to update the positions of the predators.

$$\text{Prey} = \begin{bmatrix} X_{1,1} & \cdots & X_{1,d} \\ \vdots & \ddots & \vdots \\ X_{n,1} & \cdots & X_{n,d} \end{bmatrix} \tag{31}$$

where $X_{i,j}$ is the jth dimension of the ith prey. MPA performs its steps according to three rates depending on the prey and predator speed.

2.4.1. First step

This step is used for high search ability when the step size or movement speed is large. This is the case when the predator moves faster than the prey. If the current iteration is less than one third of the maximum iteration, the position is updated according to Eq. 32. Also, the best strategy is for the hunter not to move at all. This rule is given mathematically below.

$$\text{if } iter < \frac{1}{3} Max_{iter} \text{ then}$$

$$\begin{aligned} \overrightarrow{\text{Stepsize}}_{\iota} &= \vec{R}_B \otimes \left(\overrightarrow{\text{Elite}}_{\iota} - \vec{R}_B \, \overrightarrow{\otimes\, Prey_i} \right) i = 1,2,3 \dots n \\ \overrightarrow{\text{Prey}}_{i} &= \overrightarrow{\text{Prey}}_{i} + P \cdot \vec{R} \otimes \overrightarrow{\text{stepsize}}_{i} \end{aligned} \tag{32}$$

where $\otimes$ stands for multiplication on an element-by-element basis. R_B denotes Brownian motion and takes a random number vector. $\vec{R}_B \, \overrightarrow{\otimes\, Prey_i}$ denotes the simulation of the motion of the prey. The parameter P is equal to 0.5 and R is a vector taking random values between 0 and 1.

2.4.2. Second step

In this step, search and exploitation are of equal importance, and the prey and predator move at the same speed. Thus, search tends to turn into exploitation. Half of the individuals in the population are used for search and the other half for exploitation at this stage. The prey plays the role of exploitation and the predator the role of search. Prey adopt Lévy motion and predators adopt Brownian motion. The equations for this step are defined as follows.

$$\text{if } \frac{1}{3} * t_{max} < t_{\text{current}} < \frac{2}{3} * t_{max} \tag{33}$$

The number of iterations is between one-third and two-thirds of the maximum iteration. For the first half of the population the following equations are used.

$$\begin{aligned} \overrightarrow{\text{Stepsize}}_{\iota} &= \vec{R}_L \otimes \left(\overrightarrow{\text{Elite}}_{\iota} - \vec{R}_L \otimes \overrightarrow{\text{Prey}}_{i} \right) i = 1, \ldots, n/2 \\ \overrightarrow{\text{Prey}}_{i} &= \overrightarrow{\text{Prey}}_{i} + P \cdot \vec{R} \otimes \overrightarrow{\text{Stepsize}}_{i} \end{aligned} \tag{34}$$

where R_L is a random number vector and represents the Lévy motion. The movement of the prey is modelled by $\vec{R}_L \otimes \overrightarrow{Prey_i}$. Stepsize are small steps that lead to better exploitation. For the rest of the population the following modelling is used.

$$\begin{aligned} \overrightarrow{\text{Stepsize}}_{i} &= \vec{R}_B \otimes \left(\vec{R}_B \otimes \overrightarrow{\text{Elite}}_{\iota} - \overrightarrow{\text{Prey}}_{i} \right) i = n/2, .., n \\ \overrightarrow{\text{Prey}}_{i} &= \overrightarrow{\text{Elite}}_{i} + P.CF \otimes \overrightarrow{\text{stepsize}}_{i} \end{aligned} \tag{35}$$

$$CF = \left(1 - \frac{iter}{Max_{iter}}\right)^{\left(2 \times \frac{iter}{Max_{iter}}\right)} \tag{36}$$

where CF is the control factor representing the movement of the predator. In order not to mathematically model the Brownian motion of the predator, the elite is multiplied by R_B and the prey position is updated according to the predator’s motion.

2.4.3. Third step

This step is when the speed of the predator is higher than that of the prey and corresponds to a high exploitation ability. When the number of iterations is higher than two thirds of the maximum iteration, the position is updated by the Lévy equation. Below are the equations used.

$$\begin{aligned} &\text{if } iter > \tfrac{2}{3} Max_{iter} \text{ then} \\ &\overrightarrow{\text{Stepsize}}_{\iota} = \vec{R}_L \otimes \left(\vec{R}_L \otimes \overrightarrow{\text{Elite}}_{\iota} - \overrightarrow{\text{Prey}}_{i} \right) i = 1, \ldots, n \\ &\overrightarrow{\text{Prey}}_{i} = \overrightarrow{\text{Elite}}_{\iota i} + \text{P.CF} \otimes \overrightarrow{\text{Stepsize}}_{i} \end{aligned} \tag{37}$$

In addition, two different environmental effects, fish aggregating devices (FADs) and eddy transformation, are used to better model marine predators in the MPA. This transformation is modelled in Eq. 38.

$$\overrightarrow{Prey_i} == \begin{cases} \overrightarrow{\text{Prey}}_{i} + CF \begin{bmatrix} \vec{X}_{min} \\ +\vec{R} \otimes \begin{pmatrix} \vec{X}_{max} \\ -\vec{X}_{min} \end{pmatrix} \end{bmatrix} \otimes \vec{U} & \text{if } r \leq \text{ FADs} \\ \overrightarrow{\text{Prey}}_{i} + [\text{ FADs } (1-r) + r] \begin{pmatrix} \overrightarrow{\text{Prey}}_{r1} \\ -\overrightarrow{\text{Prey}}_{r2} \end{pmatrix} & \text{if } r > \text{ FADs} \end{cases} \tag{38}$$

FAD is equal to 0.2 and is expressed as a probabilistic effect. $\vec{U}$ is a binary vector of sequences generated by generating randomly between [0,1]. If the generated number is less than 0.2, the sequence takes the value 0, otherwise it takes the value 1. r is a number that takes a random value between [0,1]. $\vec{X}_{min}$ and $\vec{X}_{max}$ indicate the lower and upper limits of the dimensions. Finally, $r1$ and $r2$ represent the random indices of the prey matrix. MPA's pseudo code is presented Algorithm 2.

Algorithm 2 MPA

Initialise $\boldsymbol{Step}$, $\boldsymbol{P}$, $\overrightarrow{\boldsymbol{Prey_i}}$.
while $t < \boldsymbol{t_{max}}$ **do**
Calculate $\boldsymbol{f}(\overrightarrow{\boldsymbol{Prey_i}})$ for each $\overrightarrow{\boldsymbol{Prey_i}}$
Create the Elite matrix.
Memorise it.
Update CF according to Eq. 36
    for each $\overrightarrow{\boldsymbol{Prey_i}}$ **do**
        if $\left(\boldsymbol{t_{current}} < \frac{1}{3} * \boldsymbol{t_{max}}\right)$ **then**
            Position $\overrightarrow{\boldsymbol{Prey_i}}$ according to Eq. 32.
        else
            if $\left(\frac{1}{3} * \boldsymbol{t_{max}} < \boldsymbol{t}_{\text{current}} < \frac{2}{3} * \boldsymbol{t_{max}}\right)$ **then**
                if $\left(\boldsymbol{i} < \frac{1}{2} * \boldsymbol{n}\right)$ **then**
                    Position $\overrightarrow{\boldsymbol{Prey_i}}$ according to Eq. 34.
                else
                    Position $\overrightarrow{\boldsymbol{Prey_i}}$ according to Eq. 35.
                end if
            else
                Position $\overrightarrow{\boldsymbol{Prey_i}}$ according to Eq. 37.
            end if
        end if
        Calculate $\boldsymbol{f}(\overrightarrow{\boldsymbol{Prey_i}})$ for each $\overrightarrow{\boldsymbol{Prey_i}}$.
        Update the position and fitness value of the best $\overrightarrow{\boldsymbol{Prey_i}}$.
        Memorize the best $\overrightarrow{\boldsymbol{Prey_i}}$.
        Apply the $\boldsymbol{FAD}$ effect according to Eq. 38.
    end for
    $\boldsymbol{t_{current}}++$
end while

The ability to recall the best places where the predator has been successful in searching for prey is related to the so-called environmental response, the so-called marine memory. In MPA, this ability is used to save memory. Once the effect of the FADs has been achieved and the location of the prey has been updated, the matrix is evaluated to increase the Elite. In the current iteration, after comparing the fitness value of each solution with the value from the previous iteration, the best one is selected. This greedy selection aims to improve the quality of the solution during each iteration. In addition, it helps hunters to remember the previous locations of the hunting areas with a successful search.

2.5. Detailed Comparison of Algorithms

The features of the algorithms are compared in detail and shown in Table 1. The algorithms have the differences listed in the table, which are due to their special features.

Table 1. Detailed comparisons of algorithms

Feature	AHA	AVOA	COA	MPA
Inspiration	Hummingbirds' flight skills and foraging strategies	African vultures' foraging and social behavior	Crayfish's summer resort, competition, and foraging behaviors	Foraging strategies of marine predators
Key Mechanisms	Three foraging strategies (guided, territorial, migration), visit table	Coefficient vector (F), phase shift mechanism, rotational search	Temperature regulation, three behavioral stages	Lévy and Brownian movements, three optimization phases
Exploration Techniques	Guided foraging, migration foraging	Rotational search equation, phase shift for diverse random motions	Temperature-based behavior regulation	Lévy flights for global search
Exploitation Techniques	Territorial foraging, memory update mechanism	Utilization of the second-best solution, four exploitation mechanisms	Local search within territorial foraging	Brownian motions for local refinement
Memory Management	Visit table to track nectar sources and visit frequency	None	None	None
Adaptive Mechanisms	Adaptive step sizes, flight pattern adjustments	Adaptive coefficients	Adaptive behaviors based on temperature	Adaptive step sizes
Computational Complexity	Low	Low	Low	Low
Unique Features	Unique flight skills simulation (axial, diagonal, omnidirectional)	Phase shift mechanism to prevent local optima trapping	Multi-stage foraging mechanism, temperature regulation for phase control	Three distinct optimization phases, combination of Lévy and Brownian motion

3. Experimental Design

In this section, the test functions to be used in performance analysis and the parameters used in the algorithms are explained.

3.1. Test functions

In this article, 26 test functions are used to evaluate the algorithms' performance. F_1 to F_{13} are unimodal functions and are given in Table 2. Likewise, the rest are multimodal functions and are presented in Table 3.

In the tables, the dimensions of F_{12} and F_{13} for unimodal functions and F_{25} and F_{26} for multimodal functions are taken as 2. For all other functions, the dimension value is set to 50. In addition, F_{min} is the minimum value that a function can reach, and the range is both the upper and lower bounds within the search space. In addition, plot graphs of some functions are shown in Figure 3.

Table 2. Unimodal test functions

Unimodal function	Range	Dimension	F_{min}
F_1 - Sphere	[-100, 100]	50	0
F_2 - Powell Sum	[-1, 1]	50	0
F_3 - Schwefel's 2.20	[-100, 100]	50	0
F_4 - Schwefel's 2.21	[-100, 100]	50	0
F_5 - Step	[-100, 100]	50	0
F_6 - Quartic Noise	[-1.28, 1.28]	50	0
F_7 - Rosenbrock	[-30, 30]	50	0
F_8 - Brown	[-1, 4]	50	0
F_9 - Dixon and Price	[-10, 10]	50	0
F_{10} - Powell Singular	[-4, 5]	50	0
F_{11} - Zakharov	[-5, 10]	50	0
F_{12} - Booth	[-10, 10]	2	0
F_{13} - Brent	[-10, 10]	2	0

Table 3. Multimodal test functions

Multimodal function	Range	Dimension	F_{min}
F_{14} - Schewel's 2.26	[-500, 500]	50	0
F_{15} - Rastrigin	[-5.12,5.12]	50	0
F_{16} - Qing	[-500, 500]	50	0
F_{17} - Alpine N. 1	[-10, 10]	50	0
F_{18} - Xin-She Yang	[-5, 5]	50	0
F_{19} - Ackley	[-32, 32]	50	0
F_{20} - Salomon	[-100, 100]	50	0
F_{21} - Griewank	[-100, 100]	50	0
F_{22} - Periodic	[-10, 10]	50	0.9
F_{23} - Xin-She Yang N.2	[-2pi, 2pi]	50	0
F_{24} - Adjiman	[-1, 2]	2	-2.02181
F_{25} - Bird	[-2pi, 2pi]	2	-106.7645
F_{26} - Egg crate	[-5, 5]	2	0

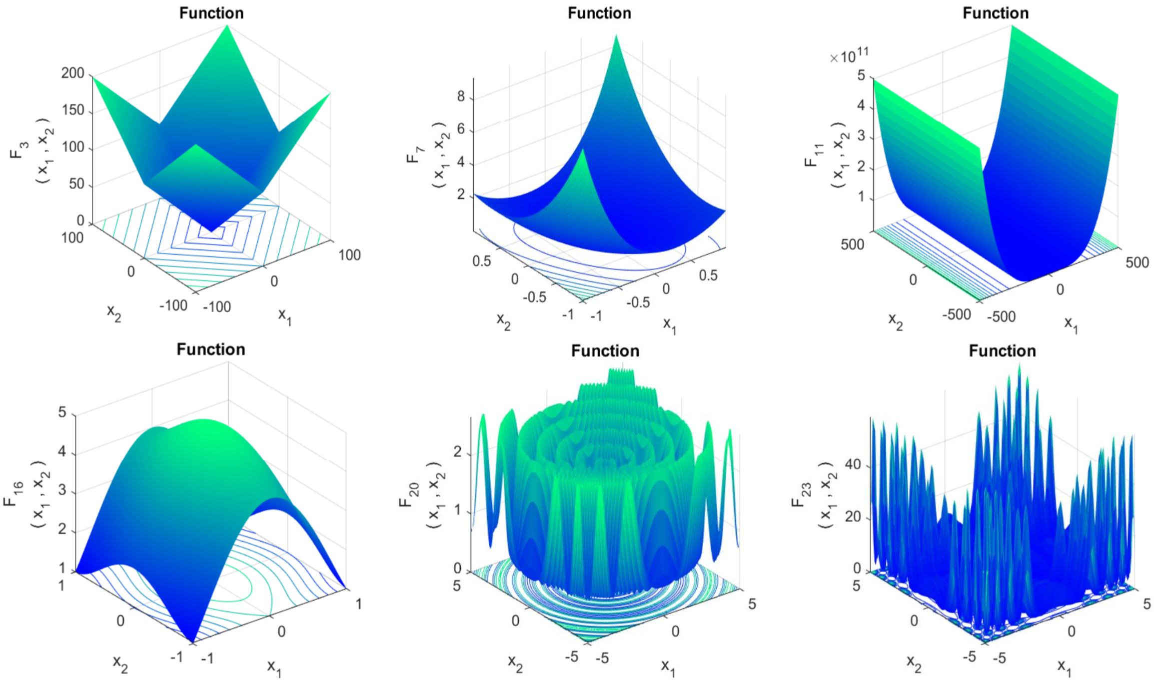

Figure 3. Plot graphs of some test functions

3.2. Parameters

Each algorithm specific parameter is given in Table 4. The population size and the maximum iteration were taken the same for all of them. Descriptions of the parameters are also given in the table.

Table 4. Parameters of algorithms

Algorithm	Parameter	Value	Description
AHA	Migration coefficient	$2n$	Migration coefficient depending on population size.
AVOA	p_1 p_2 p_3	0.6 0.4 0.6	Exploration and exploitation are the parameters for the choice of strategy.
	L_1	0.8	It gives the probability parameters for selecting the first- and
	L_2	0.2	second-best vulture.
	w	2.5	Parameter whether to terminate exploration or exploitation.
COA	C_1	0.2	It is used to control the intake of crayfish at different
	σ	3	temperatures.
	μ	25	Refers to the optimum temperature for crayfish.
	C_3	3	Explains the food factor parameter.
MPA	FADs	0.2	It is the impact factor parameter.
	P	0.5	It is a constant parameter.
All Algorithms	Population size		50
	Max. iterations		1000

5. Simulation Results

In this paper, the algorithms run 30 times independently for comparison. The simulations were carried out on MATLAB R2022b platform on an i5-12450H machine with 2.5Ghz speed and 8 GB RAM. The simulation results of unimodal and multimodal functions are given in Table 5 and Table 6, respectively. The algorithms with the lowest average and best values are presented in bold.

Simulation results evaluations for unimodal functions are as follows:

- Considering the technical characteristics of functions and algorithms in general, the following conclusions can be drawn. F_1 is a simple continuous function. It behaves the same in all dimensions. Therefore, it is easy to optimize and all functions except MPA obtained the optimum value. MPA generated results very close to the optimum, which may be due to the effect of Lévy and Brownian motions, local minima avoidance strategies and simulation of environmental effects.
- F_2 is a function that evaluates the contribution of each dimension separately. For an algorithm to be successful in this function, it needs to have adaptive parameter tuning and strong global search capabilities. Since all the algorithms we used in the comparison successfully optimize F_2, it can be said that they have these capabilities.

Table 5. Simulation results for unimodal functions

Unimodal functions	Algorithms	AHA	AVOA	COA	MPA
F_1	**Best**	**0.00E+00**	**0.00E+00**	**0.00E+00**	3.16E-48
	Worst	1.56E-292	0.00E+00	0.00E+00	4.36E-45
	Average	5.22E-294	**0.00E+00**	**0.00E+00**	7.47E-46
	Std	0.00E+00	0.00E+00	0.00E+00	1.04E-45
F_2	**Best**	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.13E-133
	Worst	0.00E+00	0.00E+00	0.00E+00	6.34E-118
	Average	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.16E-119
	Std	0.00E+00	0.00E+00	0.00E+00	1.16E-118
F_3	**Best**	5.23E-171	**0.00E+00**	**0.00E+00**	2.22E-26
	Worst	1.58E-148	0.00E+00	0.00E+00	3.49E-24
	Average	5.41E-150	**0.00E+00**	**0.00E+00**	6.14E-25
	Std	2.88E-149	0.00E+00	0.00E+00	8.35E-25
F_4	**Best**	3.01E-151	**0.00E+00**	**0.00E+00**	1.76E-18
	Worst	4.03E-134	0.00E+00	0.00E+00	3.27E-17
	Average	1.35E-135	**0.00E+00**	**0.00E+00**	1.21E-17
	Std	7.36E-135	0.00E+00	0.00E+00	7.68E-18
F_5	**Best**	**0.00E+00**	4.95E-09	2.73E-04	1.18E-08
	Worst	0.00E+00	4.42E-08	9.78E-01	6.99E-08
	Average	**0.00E+00**	1.57E-08	2.10E-01	2.87E-08
	Std	0.00E+00	8.11E-09	2.70E-01	1.44E-08
F_6	**Best**	3.06E-06	2.78E-06	**1.44E-08**	2.03E-04
	Worst	2.11E-04	1.71E-04	1.02E-04	1.46E-03
	Average	8.18E-05	3.91E-05	**2.70E-05**	6.53E-04
	Std	5.44E-05	3.90E-05	2.90E-05	2.97E-04
F_7	**Best**	5.86E-286	**0.00E+00**	**0.00E+00**	1.10E-05
	Worst	8.36E-248	0.00E+00	0.00E+00	2.41E-04
	Average	2.80E-249	**0.00E+00**	**0.00E+00**	6.58E-05
	Std	0.00E+00	0.00E+00	0.00E+00	4.95E-05
F_8	**Best**	**0.00E+00**	3.95E-27	6.34E-18	**0.00E+00**
	Worst	0.00E+00	1.38E-20	4.55E-12	0.00E+00
	Average	**0.00E+00**	1.07E-21	4.04E-13	**0.00E+00**
	Std	0.00E+00	2.98E-21	9.27E-13	0.00E+00
F_9	**Best**	4.50E+01	**9.13E-08**	4.39E+01	4.25E+01
	Worst	4.70E+01	1.45E-05	4.63E+01	4.47E+01
	Average	4.55E+01	**4.32E-06**	4.48E+01	4.33E+01
	Std	3.94E-01	4.41E-06	6.52E-01	5.29E-01

Table 5. Simulation results for unimodal functions (continued)

F_{10}	**Best**	0.00E+00	**0.00E+00**	0.00E+00	3.81E-50
	Worst	3.58E-301	0.00E+00	0.00E+00	7.12E-48
	Average	1.47E-302	**0.00E+00**	0.00E+00	1.55E-48
	Std	0.00E+00	0.00E+00	0.00E+00	1.90E-48
F_{11}	**Best**	6.67E-01	**4.48E-03**	6.67E-01	6.67E-01
	Worst	6.67E-01	5.30E-02	6.67E-01	6.67E-01
	Average	6.67E-01	**1.73E-02**	6.67E-01	6.67E-01
	Std	6.52E-17	1.01E-02	1.35E-07	3.43E-09
F_{12}	**Best**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.11E-48
	Worst	1.97E-292	0.00E+00	0.00E+00	4.46E-30
	Average	6.57E-294	**0.00E+00**	**0.00E+00**	1.49E-31
	Std	0.00E+00	0.00E+00	0.00E+00	8.14E-31
F_{13}	**Best**	**1.38E-87**	**1.38E-87**	**1.38E-87**	**1.38E-87**
	Worst	1.38E-87	1.38E-87	1.38E-87	1.38E-87
	Average	**1.38E-87**	**1.38E-87**	**1.38E-87**	**1.38E-87**
	Std	6.8E-103	6.8E-103	6.8E-103	6.8E-103

- In the functions where AHA could not achieve the optimum value, it performed very close to the other algorithms. For example, F_3 shows a very small difference of approximately 5.23E-171 compared to the others except MPA.
- In F_3 and F_4 functions, AVOA and COA's global search capability and ability to avoid local minima in a large search area made it successful in this function.
- F_5 has a flat and stepped structure and divides the solution space with sharp changes. Therefore, it can be interpreted that the AHA that is successful in this function has the ability to sample the solution space well.
- F_6 is a high-order polynomial function with a random noise term added. This function tests how robust the algorithms are to random noise. It can be evaluated that COA is closer to the optimum value (1.44E-08) than other algorithms with a noise-resistant and stable optimization capability.
- F_7 has a narrow and oblique solution path that makes it difficult to reach the global optimum. This function tests the algorithms' ability to avoid local minima while reaching the global optimum. It can be said that AVOA and COA, which are successful in this function, are due to their high routing ability and strong local search capabilities.

- The structure of F_8 tests the performance of algorithms, especially in situations that require parameter adaptations. All of the algorithms used in the comparisons have achieved success in this function with their adaptive and flexible parameterization capabilities.
- Unlike the other functions, F_9 has a significantly higher performance than the other algorithms of AVOA. This function has a large number of peaks with local minima. This means that AVOA has a large search capacity to reach the global optimum and the ability to avoid local minima.
- F_{10} has a complex structure with linear and nonlinear components. All algorithms were seen to be successful in this function. Thus, it is considered that it captures the relationships in the non-linear structure well.
- F_{11} has a complex and nonlinear structure and global optimization is difficult. AVOA, with its lower optimal value, accurately analyzes complex nonlinear relationships compared to the others.
- F_{12} and F_{13} are functions with an easy structure to optimize. All algorithms were successful with these functions. In F_{13}, all algorithms had the same success by obtaining the value 1.38E-87.
- In seven functions (F_1, F_2, F_3, F_4, F_7, F_{10} and F_{12}) AVOA and COA reached the optimum value.

Table 6. Simulation results for multimodal functions

Multimodal functions	Algorithms	AHA	AVOA	COA	MPA
F_{14}	**Best**	7.62E+00	**1.27E-05**	9.67E+01	7.55E+01
	Worst	4.05E+01	4.72E+01	1.58E+02	1.32E+02
	Average	2.24E+01	**6.03E+00**	1.32E+02	1.01E+02
	Std	8.79E+00	1.20E+01	1.45E+01	1.38E+01
F_{15}	**Best**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Worst	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Average	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Std	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F_{16}	**Best**	8.91E-01	**1.89E-03**	9.70E+01	1.29E+00
	Worst	7.14E+00	1.14E+02	6.32E+03	4.08E+02
	Average	**3.64E+00**	5.49E+00	2.14E+03	6.19E+01
	Std	1.70E+00	2.11E+01	1.56E+03	1.06E+02
F_{17}	**Best**	7.63E-167	**0.00E+00**	**0.00E+00**	7.21E-30
	Worst	6.01E-152	0.00E+00	0.00E+00	1.90E-26
	Average	2.09E-153	**0.00E+00**	**0.00E+00**	4.57E-27
	Std	1.10E-152	0.00E+00	0.00E+00	5.13E-27

Table 6. Simulation results for multimodal functions (continued)

F_{18}	**Best**	5.09E-243	**0.00E+00**	**0.00E+00**	3.65E-39
	Worst	3.81E-223	1.85E-263	0.00E+00	1.13E-23
	Average	1.29E-224	6.16E-265	**0.00E+00**	3.82E-25
	Std	0.00E+00	0.00E+00	0.00E+00	2.06E-24
F_{19}	**Best**	**-4.44E-16**	**-4.44E-16**	**-4.44E-16**	3.11E-15
	Worst	-4.44E-16	-4.44E-16	-4.44E-16	3.11E-15
	Average	**-4.44E-16**	**-4.44E-16**	**-4.44E-16**	3.11E-15
	Std	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F_{20}	**Best**	**0.00E+00**	**0.00E+00**	**0.00E+00**	9.99E-02
	Worst	1.18E-147	0.00E+00	0.00E+00	2.00E-01
	Average	**4.11E-149**	**0.00E+00**	**0.00E+00**	**1.47E-01**
	Std	2.15E-148	0.00E+00	0.00E+00	5.07E-02
F_{21}	**Best**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Worst	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Average	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Std	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F_{22}	**Best**	**9.00E-01**	**9.00E-01**	**9.00E-01**	**9.00E-01**
	Worst	9.00E-01	9.00E-01	9.00E-01	1.03E+00
	Average	**9.00E-01**	**9.00E-01**	**9.00E-01**	1.00E+00
	Std	4.52E-16	4.52E-16	4.52E-16	2.12E-02
F_{23}	**Best**	**1.21E-20**	**1.21E-20**	6.25E-20	**1.21E-20**
	Worst	3.23E-20	1.21E-20	4.72E-16	3.45E-20
	Average	1.35E-20	**1.21E-20**	3.39E-17	1.67E-20
	Std	5.09E-21	5.25E-27	9.22E-17	6.54E-21
F_{24}	**Best**	**-2.02E+00**	**-2.02E+00**	**-2.02E+00**	**-2.02E+00**
	Worst	-2.02E+00	-2.02E+00	-2.02E+00	-2.02E+00
	Average	**-2.02E+00**	**-2.02E+00**	**-2.02E+00**	**-2.02E+00**
	Std	1.36E-15	1.09E-15	1.36E-15	1.36E-15
F_{25}	**Best**	**-1.07E+02**	**-1.07E+02**	**-1.07E+02**	**-1.07E+02**
	Worst	-1.07E+02	-1.07E+02	-8.73E+01	-1.07E+02
	Average	**-1.07E+02**	**-1.07E+02**	**-1.06E+02**	**-1.07E+02**
	Std	3.09E-14	3.10E-14	3.55E+00	2.95E-14
F_{26}	**Best**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.17E-241
	Worst	0.00E+00	0.00E+00	0.00E+00	1.33E-192
	Average	**0.00E+00**	**0.00E+00**	**0.00E+00**	4.85E-194
	Std	0.00E+00	0.00E+00	0.00E+00	0.00E+00

- Similarly, AHA produced the optimum value in six functions (F_1, F_2, F_5, F_8, F_{10} and F_{12}) In addition, MPA is another algorithm that reaches the optimum value in F_8.
- In general, AVOA and COA had better performance, while AHA produced results close to these two algorithms with very small differences. MPA was behind the other algorithms except F_8 and F_{13}.

Simulation results evaluations for multimodal functions are as follows:

- As in F_9, F_{14} and F_{16} also have multiple local minima, making optimization difficult. The superiority of AVOA over other algorithms in these functions is due to its effective local minimum avoidance strategies.
- F_{17} has a complex and fluctuating solution space that is difficult to optimize. F_{18} tests how algorithms perform in a large solution space. F_{19} tests their ability to efficiently explore the large solution space. All algorithms were successful in all three functions.
- For eight functions (F_{15}, F_{17}, F_{18}, F_{20}, F_{21}, F_{24}, F_{25} and F_{26}) AVOA and COA produced optimum values.
- AHA has the same performance in all functions except F_{17} and F_{18}, where AVOA and COA reach the optimum value.
- In F_{20}, F_{21}, F_{22}, F_{23}, F_{24}, F_{25} and F_{26} functions that evaluate global search capabilities, all algorithms performed well by reaching similar objective function values.
- MPA has an optimum result in F_{15}, F_{21}, F_{24}.
- In F_{14}, no algorithm produced the minimum value, but AVOA had the best performance with the best value of 1.27E-05.
- The functions for which all algorithms produce the same value are F_{19} and F_{22}.
- Like the unimodal functions, AHA follows AVOA and COA with very small differences. At the same time, MPA was the most unsuccessful except for a few functions.
- In general, the algorithms outperformed each other with very small differences such as 7.63E-167 and 7.21E-30.

When all evaluations are taken into account, the rankings of the algorithms in terms of average and best values for all simulation results are given in Table 7.

Table 7. Rank ordering of algorithms

Algorithms	**Best**	**Average**
AHA	16	12
AVOA	**23**	**20**
COA	19	19
MPA	8	6

In Table 7, the values of the most successful algorithms are bolded. AVOA was the most successful in terms of best and average values. AVOA is followed by COA, AHA, MPA respectively. AVOA outperformed its closest competitor COA by four points in the best values and one point in the average values.

5.1. Convergence plots of test functions

A convergence plot is a type of graph often used to observe or evaluate the performance of an optimization algorithm (Yiğit et al., 2023). This graph is a visualization of how fast or effective an algorithm is in the optimization process when it tries to minimize the value of a target function. For this purpose, the convergence graphs of the averages of all runs of the algorithms on the test functions are given in Figure 5.

In the convergence plot of unimodal functions, COA was the fastest converging algorithm for most functions except for a few functions (F_5, F_8, F_9 and F_{11}). COA is followed by AVOA. At the same time, these two algorithms started to produce the same value after a certain number of iterations. In addition, AHA was the algorithm with the fastest convergence only in F_8, followed by MPA. Although F_{13} functions produced the same result, AHA reached this value later than the other algorithms.

When the convergence plots of the multidimensional functions are analysed, it is seen that COA converges faster in general. In F_{14}, although none of the algorithms produced the optimum value, AVOA converged the fastest. In F_{23}, contrary to the general results, COA is worse. Looking at the simulation results tables, it is observed that MPA generally performs poorly and is successful in some functions. This shows that MPA converges less than the others in the convergence plots.

5.2. t-test Results

The t-test is a hypothesis test used to assess statistical significance between two groups (Kim, 2015). This test is used to determine whether the difference between the means of the groups is indeed significant. In this subsection, a one-tailed t-test analysis of the simulation results was performed. The significance level in the analysis was set at 5%. An h value equal to 1 and p values less than 0.05 indicate that the algorithm on the right performs better and there is a significant difference. In addition, no value indicates that the two algorithms have the same performance on that function. The results are presented in Table 8 and Table 9 for unimodal and multimodal functions, respectively.

According to Table 8, the evaluations are as follows:

- COA significantly outperformed AVOA in F_5, F_8, F_9 and F_{10} functions.
- In seven out of thirteen functions (F_1, F_6, F_7, F_9, F_{10}, F_{11} and F_{12}), AHA is less successful, and in nine functions (F_1, F_3, F_4, F_5, F_6, F_7, F_9, F_{10} and F_{11}), MPA is less successful than AVOA.

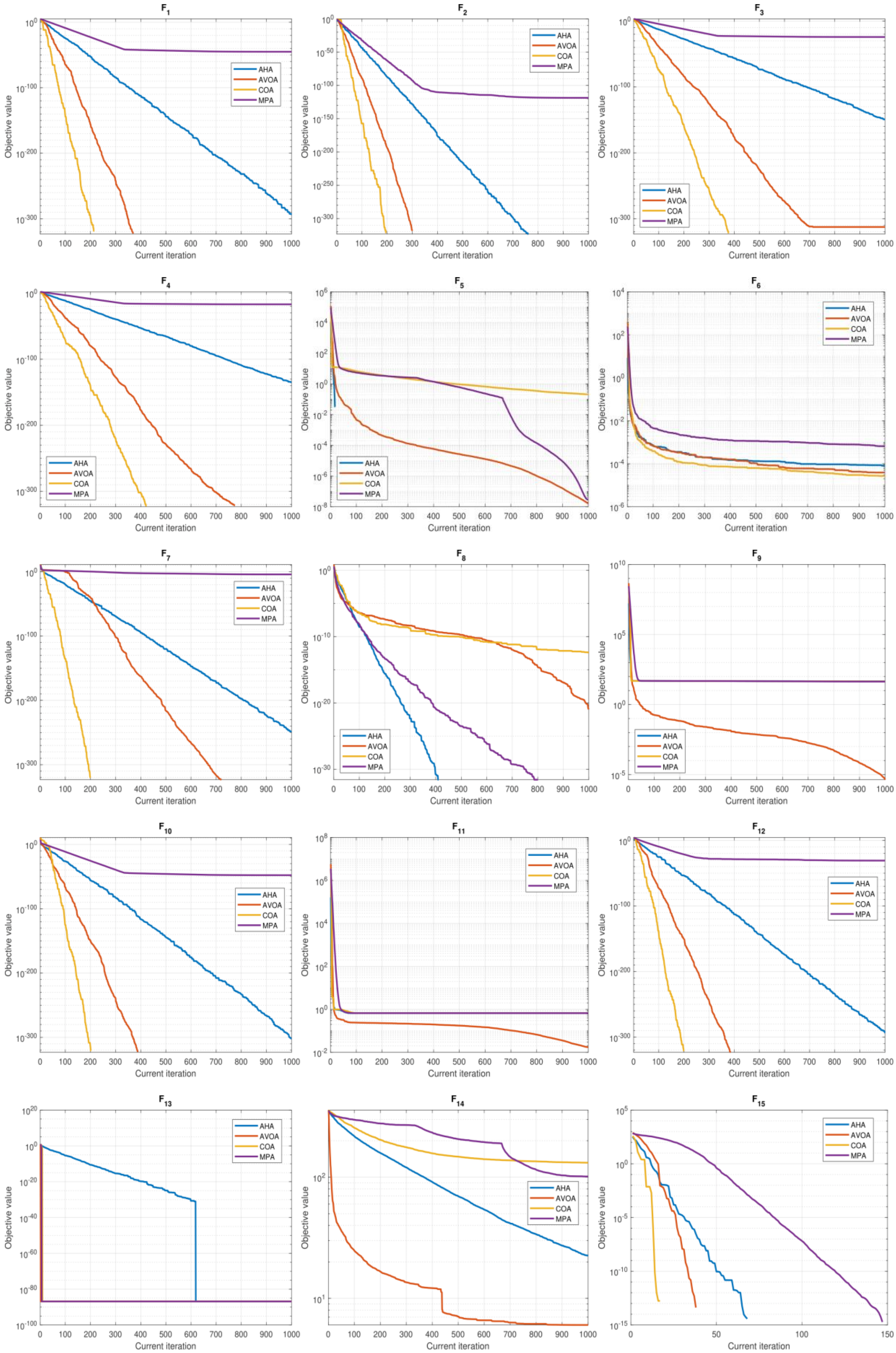

Figure 5. Convergence plots of test functions

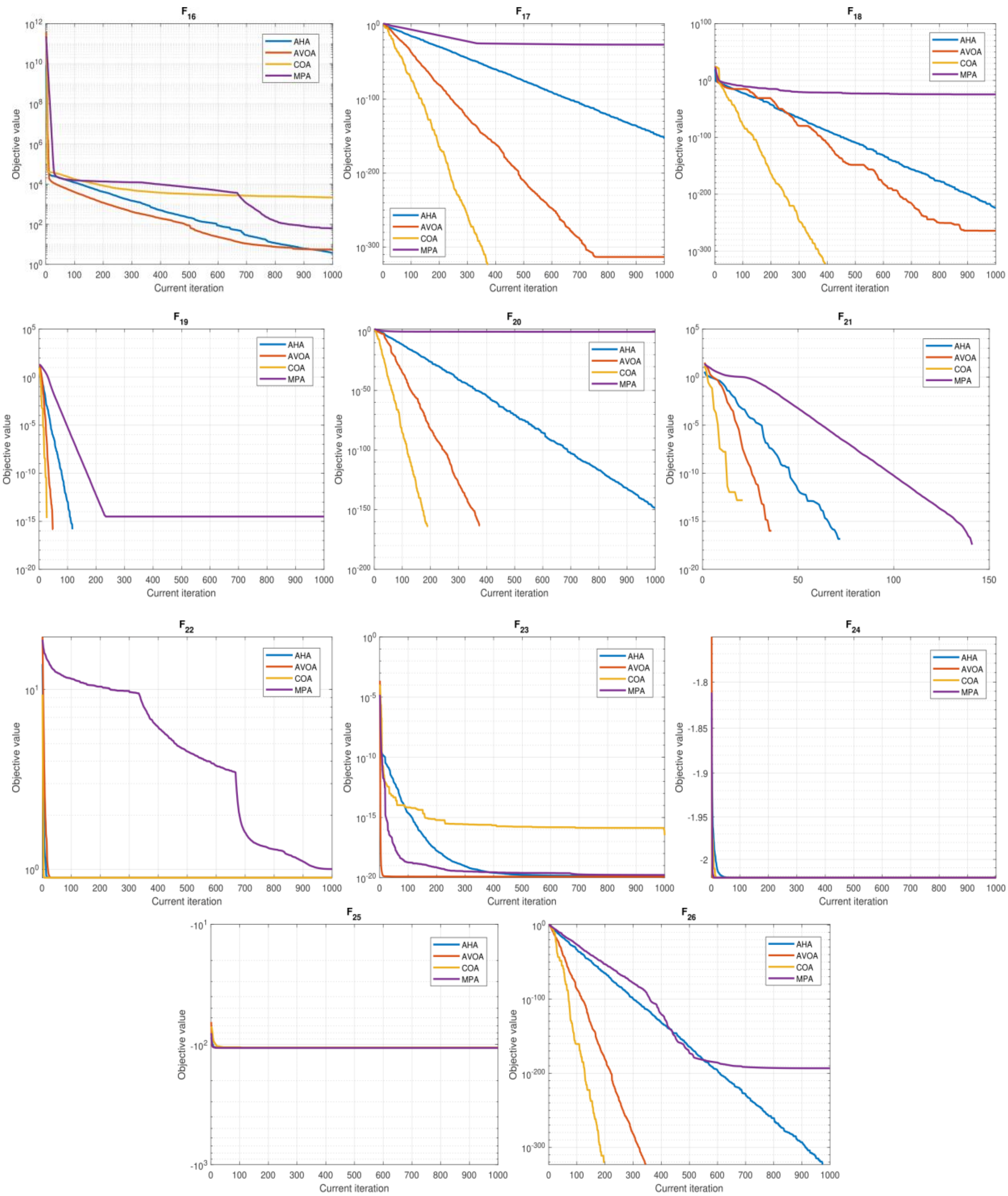

Figure 5. Convergence plots of test functions (continued)

- AHA compared to MPA in seven functions (F_1,F_4, F_5, F_6, F_7, F_{10} and F_{11}) showed that there was a significant difference.
- COA has achieved superiority in AHA (F_1,F_6, F_7, F_9, F_{10} and F_{12}) and MPA (F_1,F_4, F_5, F_6, F_7, F_{10} and F_{11}) with six functions.
- Since all algorithms produced the same value in F_{13}, no significant differences were observed.

Table 8. Unimodal functions t-test results

	AVOA-COA		AVOA-AHA		AVOA-MPA		COA-AHA		COA-MPA		AHA-MPA	
	h	p	h	p	h	p	h	p	h	p	h	p
F_1	-	-	**1.00E+00**	0.00E+00	**1.00E+00**	9.19E-05	**1.00E+00**	0.00E+00	**1.00E+00**	9.19E-05	**1.00E+00**	9.19E-05
F_2	-	-	-	-	0.00E+00	1.57E-01	-	-	0.00E+00	1.57E-01	0.00E+00	1.57E-01
F_3	-	-	0.00E+00	1.56E-01	**1.00E+00**	1.87E-04	0.00E+00	1.56E-01	**1.00E+00**	1.87E-04	0.00E+00	1.87E-04
F_4	-	-	0.00E+00	1.62E-01	**1.00E+00**	8.73E-10	0.00E+00	1.62E-01	**1.00E+00**	8.73E-10	**1.00E+00**	8.73E-10
F_5	**1.00E+00**	9.92E-05	0.00E+00	1.00E+00	**1.00E+00**	1.16E-04	0.00E+00	1.00E+00	0.00E+00	1.00E+00	**1.00E+00**	4.27E-12
F_6	0.00E+00	9.15E-01	**1.00E+00**	1.26E-03	**1.00E+00**	3.75E-12	**1.00E+00**	4.14E-05	**1.00E+00**	8.71E-13	**1.00E+00**	1.45E-11
F_7	-	-	**1.00E+00**	0.00E+00	**1.00E+00**	2.59E-08	**1.00E+00**	0.00E+00	**1.00E+00**	2.59E-08	**1.00E+00**	2.59E-08
F_8	**1.00E+00**	1.19E-02	0.00E+00	9.71E-01	0.00E+00	9.71E-01	0.00E+00	9.88E-01	0.00E+00	9.88E-01	-	-
F_9	**1.00E+00**	2.46E-55	**1.00E+00**	6.96E-62	**1.00E+00**	1.49E-57	**1.00E+00**	1.14E-05	0.00E+00	1.00E+00	0.00E+00	1.00E+00
F_{10}	-	-	**1.00E+00**	0.00E+00	**1.00E+00**	5.40E-05	**1.00E+00**	0.00E+00	**1.00E+00**	5.40E-05	**1.00E+00**	5.40E-05
F_{11}	**1.00E+00**	1.48E-54	**1.00E+00**	1.48E-54	**1.00E+00**	1.48E-54	0.00E+00	1.00E+00	0.00E+00	1.00E+00	**1.00E+00**	3.57E-14
F_{12}	-	-	**1.00E+00**	0.00E+00	0.00E+00	1.63E-01	**1.00E+00**	0.00E+00	0.00E+00	1.63E-01	0.00E+00	1.63E-01
F_{13}	-	-	-	-	-	-	-	-	-	-	-	-

Table 9. Multimodal functions t-test results

	AVOA-COA		AVOA-AHA		AVOA-MPA		COA-AHA		COA-MPA		AHA-MPA	
	h	p	h	p	h	p	h	p	h	p	h	p
F_{14}	**1.00E+00**	1.01E-24	**1.00E+00**	2.58E-06	**1.00E+00**	5.12E-22	0.00E+00	1.00E+00	0.00E+00	1.00E+00	**1.00E+00**	8.43E-23
F_{15}	-	-	-	-	-	-	-	-	-	-	-	-
F_{16}	**1.00E+00**	1.56E-08	0.00E+00	6.82E-01	**1.00E+00**	3.71E-03	0.00E+00	1.00E+00	0.00E+00	1.00E+00	**1.00E+00**	2.57E-03
F_{17}	-	-	0.00E+00	1.52E-01	**1.00E+00**	1.74E-05	0.00E+00	1.52E-01	**1.00E+00**	1.74E-05	**1.00E+00**	1.74E-05
F_{18}	0.00E+0	1.00E+0	**1.00E+00**	0.00E+00	0.00E+00	1.59E-01	**1.00E+00**	0.00E+00	0.00E+00	1.59E-01	0.00E+00	1.59E-01
F_{19}	-	-	-	-	**1.00E+00**	0.00E+0	-	-	**1.00E+00**	0.00E+00	**1.00E+00**	0.00E+00
F_{20}	**-**	-	0.00E+00	1.52E-01	**1.00E+00**	4.24E-16	0.00E+00	1.52E-01	**1.00E+00**	4.24E-16	**1.00E+00**	4.24E-16
F_{21}	-	-	-	-	-	-	-	-	-	-	-	-
F_{22}	-	-	-	-	**1.00E+00**	1.80E-22	-	-	**1.00E+00**	1.80E-22	**1.00E+00**	1.80E-22
F_{23}	**1.00E+00**	2.69E-02	0.00E+00	6.43E-02	**1.00E+00**	2.86E-04	0.00E+00	9.73E-01	0.00E+00	9.73E-01	**1.00E+00**	2.94E-02
F_{24}	0.00E+00	1.00E+0	0.00E+00	1.00E+00	0.00E+00	1.00E+0	-	-	-	-	-	-
F_{25}	0.00E+00	1.63E-01	0.00E+00	1.00E+00	0.00E+00	4.28E-01	0.00E+0	8.37E-01	0.00E+00	8.37E-01	**1.00E+00**	2.62E-10
F_{26}	-	-	-	-	**1.00E+00**	0.00E+0	-	-	**1.00E+00**	0.00E+00	**1.00E+00**	0.00E+00

According to Table 9, MPA was the algorithm that performed less well compared to the others, as in the case of unimodal functions. AVOA outperformed COA in F_{14} ,F_{16} and F_{23} functions. Similar to Table 7, AHA exhibits a more significant difference than MPA and has demonstrated superior performance in nine different test functions (F_{14}, F_{16}, F_{17}, F_{19}, F_{20}, F_{22}, F_{23}, F_{25} and F_{26}).

6. Conclusion

The development of many optimization algorithms provides different possibilities for solving problems. For this purpose, recently proposed AHA, AVOA, COA and MPA algorithms were compared for the first time and the best performance was analyzed. According to the analyses obtained from the simulation results, AVOA has the highest performance in 26 different test functions. Other newly proposed algorithms followed AVOA with very small differences. Moreover, each of them outperformed each other in some functions. The success or failure of algorithms on certain test functions compared to others can be explained by factors such as the exploration and exploitation balance of the algorithms, their working mechanisms, and the complexity and dimensionality of the test functions. The difficulty of a test function increases with the number of dimensions; therefore, the convergence of some methods is slower compared to others. For instance, algorithms with strong exploration capabilities can be successful in larger solution spaces and irregular functions, whereas algorithms effective at finding local optima can perform better on less complex functions. Additionally, the parameter settings of the algorithms and their suitability to the problem structure can significantly affect performance. The combination of these factors explains why certain algorithms perform better or worse on specific test functions. After the simulations, statistical analysis was performed to determine whether there was a significant difference. The fact that the algorithms follow each other with very small differences shows that the appropriate one should be used for different problems. In future research, it is aimed to analyze the newly proposed metaheuristics in various test functions and to use them on different problems.

Statement of Conflict of Interest

Authors have declared no conflict of interest.

Author's Contributions

The contribution of the authors is equal.

References

Abdollahzadeh B., Gharehchopogh FS., Mirjalili S. African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems. Computers & Industrial Engineering 2021; 158, 107408.

Arslan S. Güncel metasezgisel algoritmalarının performansları üzerine karşılaştırılmalı bir çalışma. Düzce Üniversitesi Bilim ve Teknoloji Dergisi 2023; 11(4): 1861-1884.

Azizi M., Talatahari S., Gandomi AH. Fire hawk optimizer: A novel metaheuristic algorithm. Artificial Intelligence Review 2023; 56(1): 287-363.

Deng L., Liu S. Snow ablation optimizer: A novel metaheuristic technique for numerical optimization and engineering design. Expert Systems with Applications 2023; 225, 120069.

Dorigo M., Birattari M., Stutzle T. Ant colony optimization. IEEE computational intelligence magazine, 2006; 1(4): 28-39.

Dowsland KA., Thompson J. Simulated annealing. Handbook of Natural Computing 2012 1623-1655.

Faramarzi A., Heidarinejad M., Mirjalili S., Gandomi AH. Marine predators algorithm: A nature-inspired metaheuristic. Expert Systems with Applications 2020; 152, 113377.

Ghaedi A., Bardsiri AK., Shahbazzadeh MJ. Cat hunting optimization algorithm: a novel optimization algorithm. Evolutionary Intelligence 2023; 16(2): 417-438.

Hosseinzadeh M., Rahmani AM., Husari FM., Alsalami OM., Marzougui M., Nguyen GN., Lee SW. A survey of artificial hummingbird algorithm and its variants: statistical analysis, performance evaluation, and structural reviewing. Archives of Computational Methods in Engineering 2024; 1-42.

Jia H., Rao H., Wen C., Mirjalili S. Crayfish optimization algorithm. Artificial Intelligence Review 2023; 56(Suppl 2): 1919-1979.

Karaboga D. Artificial bee colony algorithm. Scholarpedia 2010; 5(3): 6915.

Kennedy J., Eberhart R. Particle swarm optimization. IEEE In Proceedings of ICNN'95-International Conference on Neural Networks 1995; 4: 1942-1948.

Kim TK. T test as a parametric statistic, Korean Journal of Anesthesiology 2015; 68(6): 540-546.

Mirjalili S. Genetic algorithm. Evolutionary Algorithms and Neural Networks: Theory and Applications 2019; 43-55.

Mishra P., Singh U., Pandey CM., Mishra P., Pandey G. Application of student’s t-test, analysis of variance, and covariance. Annals of Cardiac Anaesthesia 2019; 22(4): 407.

Xue Y., Jia W., Zhao X., Pang W. An evolutionary computation based feature selection method for intrusion detection, Security and Communication Networks 2018.

Yang XS. A new metaheuristic bat-inspired algorithm. Nature inspired cooperative strategies for optimization. Berlin, Heidelberg: Springer Berlin Heidelberg 2010; 65-74.

Yang XS., Deb S. Cuckoo search via Lévy flights. IEEE World congress on nature & biologically inspired computing (NaBIC) 2009; 210-214.

Yiğit H., Ürgün S., Mirjalili S. Comparison of recent metaheuristic optimization algorithms to solve the SHE optimization problem in MLI, Neural Computing and Applications 2023; 35(10): 7369-7388.

Zhao S., Zhang T., Ma S., Chen M. Dandelion optimizer: A nature-inspired metaheuristic algorithm for engineering applications. Engineering Applications of Artificial Intelligence 2022; 114, 105075.

Zhao W., Wang L., Mirjalili S. Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications. Computer Methods in Applied Mechanics and Engineering 2022; 388, 114194.

Zhu D., Wang L., Zhou C., Yan S., Xue J. Human memory optimization algorithm: A memory-inspired optimizer for global optimization problems. Expert Systems with Applications 2023; 237, 121597.