Volume 14, Issue 3, Page 73-89, 2025 https://doi.org/10.46810/tdfd.1610740 Research Article

Türk Doğa ve Fen Dergisi
Turkish Journal of Nature and Science
www.dergipark.gov.tr/tdfd

Performance Analysis of Four Metaheuristic Algorithms on Benchmark Functions

Sibel ARSLAN[1*], Muhammed Furkan GÜL[2]

[1] Sivas Cumhuriyet University, Faculty of Engineering, Department of Software Engineering, Sivas, Türkiye
[2]Sivas University of Science and Technology, Faculty of Engineering and Natural Sciences, Department of Computer Engineering, Sivas, Türkiye
Sibel ARSLAN ORCID No: 0000-0003-3626-553X
Muhammed Furkan GÜL ORCID No: 0009-0007-0486-0525

Corresponding author: sibelarslan@cumhuriyet.edu.tr

(Received: 31.12.2024, Accepted: 15.07.2025, Online Publication: 26.09.2025)

Keywords
Metaheuristics, Artificial rabbit optimization algorithm, African vulture optimization algorithm, Prairie dog optimization algorithm, Genetic algorithm

Abstract: Various metaheuristic algorithms inspired by nature are used to solve optimization problems. With the increasing number of metaheuristics, their performance on problems is gradually improving. In this paper, the performance analysis of the newly proposed metaheuristics Artificial Rabbit Optimization Algorithm (ARO), African Vulture Optimization Algorithm (AVOA), Prairie Dog Optimization Algorithm (PDO) and the well-known Genetic Algorithm (GA) were performed for the first time. ARO is modeled after rabbits' behavioral patterns, such as detour foraging and random hiding. AVOA is developed based on the navigation and competitive behaviors of African vultures. The newly proposed final metaheuristic PDO is inspired by the survival struggle of prairie dogs. As for the popular GA, it is based on survival of the fittest. Unimodal and multimodal test functions were used during the analysis. According to the simulation results, AVOA performed better and generated more successful results compared to the others 22 times in the mean and best values. AVOA was followed by PDO and ARO, proving that the newly proposed metaheuristics will be successful on different problems.

Dört Metasezgisel Algoritmanın Kıyaslama Fonksiyonları Üzerindeki Performans Analizi

Anahtar Kelimeler
Metasezgiseller, Yapay tavşan optimizasyon algoritması, Afrika akbabası optimizasyon algoritması, Çayır köpeği optimizasyon algoritması, Genetik algoritma

Öz: Doğadan ilham alan çeşitli metasezgisel algoritmalar, optimizasyon problemlerini çözmek için kullanılmaktadır. Metasezgisel algoritmaların sayısındaki artışla birlikte, bu algoritmaların problemlerdeki performansları da giderek iyileşmektedir. Bu makalede, yeni önerilen metasezgisel algoritmalar olan Yapay Tavşan Optimizasyon Algoritması (ARO), Afrika Akbaba Optimizasyon Algoritması (AVOA), Çayır Köpeği Optimizasyon Algoritması (PDO) ve iyi bilinen Genetik Algoritma'nın (GA) performans analizleri ilk kez gerçekleştirilmiştir. ARO, tavşanların dolambaçlı beslenme ve rastgele saklanma gibi davranış kalıplarını model alarak geliştirilmiştir. AVOA, Afrika akbabalarının navigasyon ve rekabetçi davranışlarına dayanmaktadır. Yeni önerilen son metasezgisel algoritma PDO ise çayır köpeklerinin hayatta kalma mücadelesinden esinlenilerek geliştirilmiştir. Popüler GA ise en uygun olanın hayatta kalması prensibine dayanır. Analiz sırasında tek modlu (unimodal) ve çok modlu (multimodal) test fonksiyonları kullanılmıştır. Simülasyon sonuçlarına göre, AVOA diğerlerine kıyasla 22 kez ortalama ve en iyi değerlerde daha iyi performans göstermiş ve daha başarılı sonuçlar üretmiştir. AVOA'yı PDO ve ARO takip ederek, yeni önerilen metasezgisel algoritmaların farklı problemlerde başarılı olacağını kanıtlamıştır.

1. INTRODUCTION

Nowadays, the number of metaheuristic algorithms developed inspired by nature is increasing. This increase is due to the fact that metaheuristics are less costly and more effective than traditional approaches. For this reason, they are used in many different fields, especially engineering [1,2]. Metaheuristics used in these areas contribute to the analysis of large data sets, efficient use of resources, improvement of decision-making processes and the solution of many other complex problems. Many popular metaheuristics that contribute to the solution of these complex problems such as Particle Swarm Optimization Algorithm [3], Genetic Algorithm (GA) [4], Artificial Bee Colony Algorithm [5] and Ant Colony Optimization Algorithm [6] are among them. However, newly developed metaheuristics perform better in less time compared to these popular ones [7]. With the increase in performances, determining the best metaheuristic provides important contributions to optimization problems.

The aim of this paper is to compare the performances of the newly proposed metaheuristics, namely Artificial Rabbit Optimization Algorithm (ARO), African Vultures Optimization Algorithm (AVOA), Prairie Dog Optimization Algorithm (PDO), along with the popular metaheuristic GA, using various performance criteria. The motivation for selecting ARO, AVOA, and PDO lies in their novelty and increasing presence in recent metaheuristic research. These algorithms, developed between 2021 and 2023, are inspired by diverse biological systems—rabbits, vultures, and prairie dogs—which offer a wide behavioral spectrum for optimization modeling. Despite promising initial findings in their original proposals, no comprehensive and independent comparison has been conducted under identical test environments. This study aims to address that gap and to evaluate their performance against a well-known algorithm, GA. Other contributions of the paper are as follows:

- To the best of our knowledge, although ARO, AVOA, and PDO have been individually tested in their original studies, this is the first independent work to analyze all three under identical experimental conditions and compare them directly with a common reference algorithm (GA). This provides a more objective assessment of their relative performance.
- The performances of metaheuristics are analyzed with different test functions.
- According to the information obtained from the experiments, AVOA produces the best performance.

In the remainder of the paper is as follows. Section 2 explains information about metaheuristics and their pseudocodes. Section 3 presents the test functions used to analyze the performance of metaheuristics. Additionally, in this section, the parameters of the metaheuristics are given. Section 4 provides simulation results, convergence curves, and t-test results. Finally, Section 5 gives information about the conclusion and future work.

2. MATERIAL AND METHOD

In this section, detailed information about metaheuristic algorithms is given. In order to present the algorithms in a more comprehensive and diverse manner, this study includes pseudocode representations for ARO and AVOA, and flowcharts for GA and PDO. This mixed presentation aims to enhance understanding by offering both algorithmic logic and visual summaries.

2.1. Artificial Rabbit Optimization Algorithm (ARO)

ARO is a metaheuristic based on behavioral models of rabbits [8]. In ARO, the two behaviors that rabbits have are determined according to the energy of the rabbits, and the transition between the behavior is made depending on energy shrink. Initially, the energy levels of the rabbits are high and in order to expand the search space, the rabbits exhibit detour foraging behavior. They do this by selecting grasses in remote areas to prevent predators from finding their nests. Energy shrink occurs when foraging becomes repetitive. When there is enough energy shrink, they switch to random hiding behavior. In order to update their recent position, they build many nests in their territory and aim to hide from predators. They randomly select one of the burrows and complete the random hiding. Energy shrink between detour foraging and random hiding then these two behavioral strategies are described in detail below.

Energy shrink

In the early phases of the iterations, rabbits consistently exhibit detour foraging behavior [8]. However, in the later phases of the iterations they perform random hiding. This intermediate transition is caused by energy shrink and its mathematical model is presented in Equation 1.

$$A(t) = 4\left(1 - \frac{t}{T}\right)\ln\frac{1}{r} \quad (1)$$

where r represents a randomly generated number within the range of 0 to 1. t is the recent number of iterations, while T depicts the total number of iterations.

Detour foraging

Rabbits have a wide field of vision. For this reason, they do not eat food in their habitat to avoid predators detecting their nests [9]. They randomly feed on grasses in remote areas. This behavior is termed detour foraging (exploration). In detour foraging, the ARO helps to avoid local extremes and search globally, as described in Equation 2.

$$\vec{v}_i(t+1) = \vec{x}_j(t) + R \cdot \left(\vec{x}_i(t) - \vec{x}_j(t)\right) + round\left(0.5 \cdot (0.05 + r_1)\right) \cdot n_1, \quad i,j = 1, \dots, n \text{ and } j \neq i \tag{2}$$

where $\vec{v}_i$ (t+1) represents the ith rabbit's likely position at time t+1. $\vec{x}_i$ (t) is the ith rabbit's position at time t. n is the rabbit population size. round is rounding to the nearest integer. n_1 depends on the standard normal distribution and is shown in Equation 3. r_1 is a random number from 0 to 1. R is the running operator and is expressed as in Equation 4.

$$n_1 \sim N(0,1) \tag{3}$$

$$R = L \cdot c \tag{4}$$

Equation 5 defines the run length, denoted as L, which signifies the movement speed during detour foraging. c is a mapping vector that can randomly assist the algorithm and is mathematically modeled as in Equations 6-7.

$$L = \left(e - e^{\left(\frac{t-1}{T}\right)^2}\right) \cdot \sin(2\pi r_2) \tag{5}$$

where t and T denote the recent and total number of iterations respectively. r_2 is a random number ranging between 0 and 1.

$$c(k) = \begin{cases} 1 & \text{if } k == g(l) \\ 0 & \text{else} \end{cases} \quad k = 1, \dots, d \text{ and } l = 1, \dots, \lceil r_3 \cdot d \rceil \tag{6}$$

$$g = \text{randperm}(d) \tag{7}$$

where $\lceil \cdot \rceil$ represents the ceiling function, randperm is an integer permutation from 1 to d at random. r_3 is a random number between 0 and 1.

Random hiding

Rabbits can dig different tunnels around their burrows to escape predators [10]. In each iteration, a rabbit digs d tunnels in the search space in each dimension. They also randomly choose one of the tunnels in each dimension to reduce the probability of predation. Mathematical models of random hiding are given in Equations 8-10.

$$\vec{v}_i(t+1) = \vec{x}_i(t) + R \cdot \left(r_4 \cdot \vec{b}_{i,r}(t) - \vec{x}_i(t)\right), i = 1, \dots, n \tag{8}$$

$$g_r(k) = \begin{cases} 1 & \text{if } k == \lceil r_5 \cdot d \rceil \\ 0 & \text{else} \end{cases} \quad k = 1, \dots, d \tag{9}$$

$$\vec{b}_{i,r}(t) = \vec{x}_i(t) + H \cdot g_r \cdot \vec{x}_i(t) \tag{10}$$

where $\vec{b}_{i,r}(t)$ is the randomly chosen hiding place. r_4 and r_5 are random numbers from 0 to 1. $\vec{x}_i(t)$ is the ith rabbit's position at time t. R is the running operator, H is the hiding parameter and d is the problem size. The location update after performing detour foraging or random hiding is given in Equation 11.

$$\vec{x}_i(t+1) = \begin{cases} \vec{x}_i(t) & f(\vec{x}_i(t)) \le f(\vec{v}_i(t+1)) \\ \vec{v}_i(t+1) & f(\vec{x}_i(t)) > f(\vec{v}_i(t+1)) \end{cases} \tag{11}$$

In cases where the fitness value of the candidate position of the ith rabbit exceeds its recent position, The rabbit will move from where it was and stay at the candidate place that either (Equation 2) or (Equation 8) determines. The ARO pseudocode is presented in Algorithm 1.

```
Algorithm 1: ARO
1.  Generate the initial population randomly and evaluate their
    fitness
2.  repeat
3.      for each (rabbit) do
4.          Calculate the energy shrink using (Equation 1)
5.          if (energy shrink > 1) then
6.              Choose a random rabbit
7.              Calculate the running operator using
    (Equations 3-7)
8.              Perform detour foraging using (Equation 2)
9.              Calculate fitness value
10.             Update recent individual's position using
    (Equation 11)
11.         else
12.             Generate the nests using (Equation 10)
13.             Perform random hiding using (Equation 8)
14.             Calculate fitness value
15.             Update recent individual's position using
    (Equation 11)
16.         end if
17.         Update the best solution so far
18.     end for
19. until (stopping criterion is satisfied)
20. return the best solution
```

2.2. African Vultures Optimization Algorithm (AVOA)

AVOA is a metaheuristic inspired from the navigation and competition behavior of African vultures [11]. The vultures are divided into two basic groups, each representing a solution. In the algorithm, the fitness value of all solutions is calculated to divide the vultures into groups. The best vulture in the initial group is the first vulture with the highest value. Likewise, the second vulture in the second group is the best vulture in terms of value. Other vultures in the population are used to move or replace these two best vultures.

Vultures are divided into two groups to find food and live in groups. Each group has different foraging and eating abilities. Vultures are prevented from overeating by their tendency to forage and eat for hours. In the metaheuristic, the worst solution is the hungriest and weakest vulture. The other vultures try to get closer to the best vulture by avoiding the worst solution. There are four basic phases in AVOA: Identifying the best vulture in a random group, calculation of hunger rates, exploration and exploitation.

Identifying the best vulture in a random group

The initial population is created, and the solutions' fitness values are calculated. In this phase, the best vulture of the first and second group is selected from the two best solutions, respectively. The other solutions aim to reach the best solutions by moving towards the best two groups.

During each iteration, the population's positioning is adjusted according to their fitness values.

Calculation of the hunger rates

Vultures can fly longer distances when they have high energy after eating their fill [12]. If they're hungry, it means they lack sufficient energy and cannot fly next to a stronger vulture. The hunger rate, which tends to decrease, is given in Equations 12-13.

$$t = h \times \left(\sin^{w} \left(\frac{\pi}{2} \times \frac{iter}{max_{iter}} \right) + \cos \left(\frac{\pi}{2} \times \frac{iter}{max_{iter}} \right) - 1 \right) \tag{12}$$

$$F = (2rand_1 + 1) \times z \times \left(1 - \frac{iter}{max_{iter}} \right) + t \tag{13}$$

where, iter and max_{iter} are the recent and total number of iterations respectively. w is a fixed number and decreasing this parameter reduces the probability of starting the exploration. The parameter h is a randomly selected number that can assume values ranging from -2 to 2. Similarly, $rand_1$ is a random number from 0 to 1. F is the hunger rate of vultures. According to the value of z, it is determined whether vultures are hunger or not. If this value is below 0, it indicates the vulture's hunger. If it is above 0, it means that the vulture is fed. In addition, if the hunger rate of the vultures is greater than 1, they start searching for food in different regions and perform the exploration. Otherwise, they move to the exploitation by searching for food near neighboring solutions.

Exploration

Vultures choose two different strategies by searching random areas. They choose strategies based on the parameter p_1 . This parameter should have a value between 0 and 1 and should be evaluated before exploration. The mathematical model of strategy selection is described in Equation 14.

$$P(i+1) = \begin{cases} R(i) - |X \times R(i) - P(i)| \times F & p_1 \geq rand_{P_1} \\ R(i) - F + rand_2 \times \big((u_b - l_b) \times rand_3 + l_b \big) & p_1 < rand_{P_1} \end{cases} \tag{14}$$

where, $R(i)$ is the best vultures and X is the distance the vultures move to protect the food. The rand symbols in the equation are numbers between 0 and 1. u_b and l_b are the boundaries of the search space. The convergence of $rand_3$ to 1 increases the ability to explore different spaces.

Exploitation

If the hunger rate is less than 1, it starts the stage of metaheuristic exploitation. Depending on whether this rate is less than 0.5 or not, this phase is divided into two. If the value is small, vultures compete for food. Two different strategies are selected for each choice with randomly generated values. The selection of strategies is determined by the parameters p_2 and p_3. Vultures have enough energy to search for food during the competition phase and may conflict over food sources. Weaker vultures fly in a spiral pattern and try to take food from stronger ones. This behaviour is given in Equation 15, depending on the parameter p_2.

$$P(i+1) = \begin{cases} |X \times R(i) - P(i)| \times (F + rand_4) - \big(R(i) - P(i)\big) \; if \; p_2 \geq rand_{P_2} \\ R(i) - R(i) \times \left(\frac{P(i)}{2\pi} \right) \big(rand_5 \times \cos(P(i)) + rand_6 \times \sin(P(i)) \big) \; if \; p_2 < rand_{P_2} \end{cases} \tag{15}$$

where $P(i)$ represents the recent vector position from which the vulture's distance from the best vultures in two groups is calculated. $R(i)$ denotes one of the two best vultures' position vectors in the last iteration. $rand$ values are numbers between 0 and 1.

In the two stages of the exploration, there are aggressive competitions over the food source. If the hunger rate is less than 0.5, this stage of the phase is started. At the beginning of the stage, a parameter $rand_{P_3}$ is generated between 0 and 1. If this parameter p_3 is greater than or equal, several species of vultures gather on the food source. Otherwise, there is a siege strife among the vultures. The mathematical model for this stage is given in Equation 16. Additionally, the gathering of vultures over the food source is expressed based on Equations 17-18.

$$P(i+1) = \begin{cases} Eq.18 \; if \; p_3 \geq rand_{P_3} \\ Eq.19 \; if \; p_3 < rand_{P_3} \end{cases} \tag{16}$$

$$A_1 = -F \times \frac{\text{BestVulture}_1(i) \times P(i)}{\text{BestVulture}_1(i) - P(i)^2} + \text{BestVulture}_1(\mathrm{i})$$
$$A_2 = -F \times \frac{\text{BestVulture}_2(i) \times P(i)}{\text{BestVulture}_2(i) - P(i)^2} + \text{BestVulture}_2(\mathrm{i}) \tag{17}$$

$$P(i+1) = \frac{A_1 + A_2}{2} \tag{18}$$

where, $\text{BestVulture}_1(\mathrm{i})$ and $\text{BestVulture}_2(\mathrm{i})$ represent the best vultures. F is the hunger rate, P(i) is the recent vulture's position and $\mathrm{P(i+1)}$ is the position of the vulture in the next iteration.

When the hunger rate is less than 0.5, the leader vultures of the groups remain hungry. Therefore, they lack the necessary energy to handle the other vultures within the group. However, the remaining vultures also grow more aggressive in their search for food. The leader vultures move in the right different directions. Equation 19 is used to model this movement.

$$P(i+1) = R(i) - F \times LF\,(d) \times |R(i) - P(i)| \tag{19}$$

where d is the problem size and $|R(i) - P(i)|$ is the vulture's distance from among the best vultures in the two groups. To improve the efficiency of AVOA, a Lévy flight [13, 14] is included. The modeling of this flight is given in Equation 20.

$$LF(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}}, \sigma = \left(\frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\rho}{2}\right)}{\Gamma(1+\beta 2) \times \beta \times 2\left(\frac{\beta-1}{2}\right)} \right)^{\frac{1}{\beta}} \tag{20}$$

where u and v are randomly chosen numbers within the range of 0 to 1. The default number value of β is 1.5 and is constant. The AVOA pseudocode algorithm is given in Algorithm 2.

Algorithm 2: AVOA

```
Randomly generate the initial population
repeat
    Calculate fitness values of vultures
    Set the best first position of vulture
    Set the second best position of vulture
    for each (vulture) do
        Choose the best vulture position
        Update hunger rate
        if (hunger rate ≥ 1) then
            if (P_1 ≥ rand_P1) then
                Update the position using Equation 14
            else
                Update the position using Equation 14 (part two)
            end if
        end if
        if (hunger rate < 1) then
            if (hunger rate ≥ 0.5) then
                if (P_1 ≥ rand_P2) then
                    Update the position using Equation 15
                else
                    Update the position using Equation 15 (part two)
                end if
            else
                if (P_1 ≥ rand_P2) then
                    Update the position using Equation 16
                else
                    Update the position using Equation 16 (part two)
                end if
            end if
until (stopping criterion is satisfied)
return the position of the best vulture
```

2.3. Genetic Algorithm (GA)

GA is a metaheuristic algorithm inspired by natural selection and based on survival of the fittest [4]. The algorithm generates the next generation by starting genetic variations and selection processes from a randomly selected initial population. Crossover, mutation and selection operators are used to find the best solution. Problem-specific solutions are customized and encoded as fixed bit strings. Solutions are represented by chromosomes. The first mechanism used in solution improvement, crossover is the replacement of a chromosome or chromosomes passed from generation to generation [15], as shown in Figure 1.

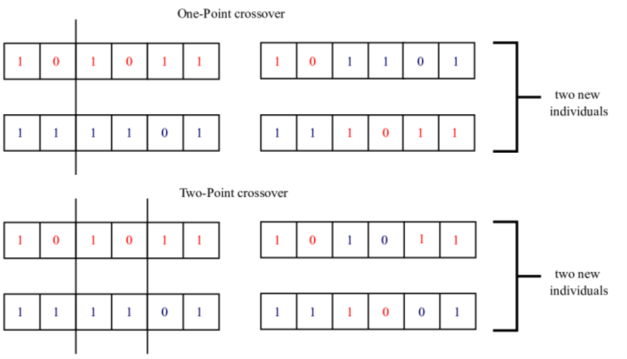

Figure 1. Crossover

In crossover, as seen in Figure 1, there are two different methods. In one-point crossover, a segment is taken from one individual at a one point, and the remaining segment is exchanged with the corresponding segment from the other individual. In two-point crossover, two segments are taken from two different points, and the chromosomes in between are swapped with those of the other individual. Another improvement mechanism, mutation is the replacement of chromosomes that give rise to a gene [16], as shown in Figure 2.

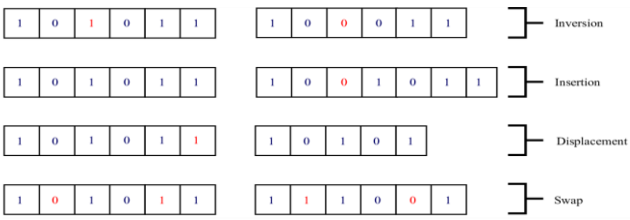

Figure 2. Mutation

Mutation can occur in four different ways: inversion, insertion, displacement, and swap. In inversion, a chromosome's value is reversed. In insertion, a new chromosome is added. Displacement occurs by removing a selected chromosome from its gene sequence. In swap, the positions of two randomly selected chromosomes are exchanged. In GA, the best individual solution is obtained when the termination criterion is met. Additionally, the GA flowchart is given in Figure 3.

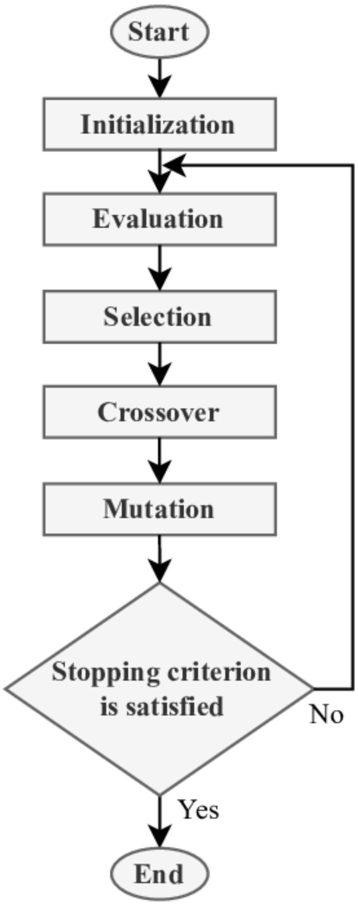

Figure 3. GA flowchart [4]

2.4. Prairie Dog Optimization Algorithm (PDO)

PDO is developed by the survival of prairie dogs and modeling their behavior [17]. In the modeling, their feeding patterns are first used to expand the problem search space. They then look for strategic positions when searching for food. These positions should serve a certain purpose that improves the general functioning of the coterie, the prairie dog community. This purpose enhances exploration, which is the search for new solutions in different regions. After the exploration, the communication skills of prairie dogs against environmental threats play an important role in their ability to prevent predation. The skills enable predators to react differently to different hunting strategies. Reactions carry out the exploitation, which aims to increase fitness to make improvements to existing solutions. The exploration and exploitation are given below.

Exploration

The first strategy for members of the coterie during the exploration is to search for new food sources in the coterie. Prairie dogs are best at catching food sources using Lévy flight movement. They communicate the precise position of food sources to other members by making distinct sounds. Once the food source quality is reached, the best one is selected for food search, and new nests are built based on the food source quality. The position update for the search in the exploration of the metaheuristic is represented in Equation 21. In addition, the Lévy flight movement is as in Equation 20.

$$\mathrm{PD}_{\mathrm{i+1,j+1}} = \mathrm{GBest}_{\mathrm{i,j}} - e\,\mathrm{CBest}_{\mathrm{i,j}} * \rho - \mathrm{CPD}_{i,j} * * \mathrm{LF}\,(n) \forall\ \mathrm{iter} < \frac{\mathrm{Max}_{\ iter}}{4} \tag{21}$$

where $PD_{i+1,j+1}$ represents the $(\mathrm{j}+1)$th dimension of the $(\mathrm{i}+1)$th prairie dog in a coterie. For this experiment, the particular food source alarm, denoted by ρ, is set at 0.1 kHz. Likewise, the second strategy involves evaluating the availability and quality of food sources and assessing the digging strength. New nests are constructed according to the digging strength, a parameter intentionally reduced with each successive iteration. This helps to limit the number of nests that can be constructed. The position update for nest building is given in Equation 22.

$$\mathrm{PD}_{\mathrm{i+1,j+1}} = \mathrm{GBest}_{\mathrm{i,j}} * \mathrm{rPD} * \mathrm{DS} * \mathrm{LF(n)} \forall \frac{\mathrm{Max}_{\ \mathrm{iter}}}{4} \leq \mathrm{iter} < \frac{\mathrm{Max}_{\ \mathrm{iter}}}{2} \tag{22}$$

where rPD is a random solution's position. When $\mathrm{GBest}_{\ \mathrm{i,j}}$ represents the globally best solution obtained so far, $e\mathrm{CBest}_{\ \mathrm{i,j}}$ evaluates the impact of the recent best solution as shown in Equation 23. DS represents the digging strength of a random value range group determined by Equation 25, which relies on food source quality. 78

$$eCBest_{i,j} = Best_{\ i,j} * \Delta + \frac{PD_{i,j} * mean\left(PD_{n,m}\right)}{\mathrm{GBest}_{\ i,j} * \left(UB_j - LB_j\right) + \Delta} \tag{23}$$

where UB_j and LB_j represent the boundaries for the jth dimension in the optimization problem., respectively. $CPD_{i,j}$ is the random cumulative effect of the whole prairie dogs in the population and is defined in Equation 24.

$$CPD_{i,j} = \frac{\mathrm{GBest}_{\ i,j} - rPD_{i,j}}{\mathrm{GBest}_{\ i,j} + \Delta} \tag{24}$$

$$DS = 1.5 \times r \times \left(1 - \frac{iter}{Max_{iter}}\right)^{\left(2\frac{\mathrm{iter}}{Max_{iter}}\right)} \tag{25}$$

where r is a parameter that takes the value -1 or 1. Δ represents a small number explaining existing differences. $iter$ and Max_{iter} represent the recent and total iteration numbers, respectively.

Exploitation

Prairie dogs use vocalizations or signals for various situations, ranging from predator dangers to availability of food [18]. Communication plays an essential part in prairie dogs' ability to meet their nutritional needs and protect against predation. They can also convey distinctions in the quality of food sources, predator presence, and hunting behaviors. Various communication skills in prairie dogs facilitate the discovery of improved or nearly ideal solutions. These solutions result in approaching a certain position or a promising position in the case of the PDO application, where more searches are conducted. The aim of the exploitation mechanisms used in PDO is to intensively explore promising areas identified during the exploration. The two strategies for this phase are Equations 26-27.

$$PD_{i+1,j+1} = \mathrm{GBest}_{i,j} - e\,\mathrm{CBest}_{i,j} * \varepsilon - CPD_{i,j} * \mathrm{rand}\ \forall \frac{\mathrm{Max}_{\mathrm{iter}}}{2} \leq \mathrm{iter} < 3\frac{\mathrm{Max}_{\mathrm{iter}}}{4} \quad (26)$$

$$PD_{i+1,j+1} = \mathrm{GBest}_{i,j} * PE * \mathrm{rand}\ \forall 3\frac{\mathrm{Max}_{\mathrm{iter}}}{4} \leq \mathrm{iter} < \mathrm{Max}_{\mathrm{iter}} \quad (27)$$

where $PD_{i+1,j+1}$ represents the $(j+1)$th dimension of the $(i+1)$th prairie dog. ε is a little number representing food source quality. PE denotes the predator effect modeled by Equation 28 and rand indicates a number between 0 and 1, which is random.

$$PE = 1.5 \times \left(1 - \frac{iter}{Max_{\mathrm{iter}}}\right)^{\left(2\frac{iter}{Max_{iter}}\right)} \quad (28)$$

where iter and Max_{iter} are the recent and total number of iterations, respectively. The flowchart of the PDO is given in Figure 4.

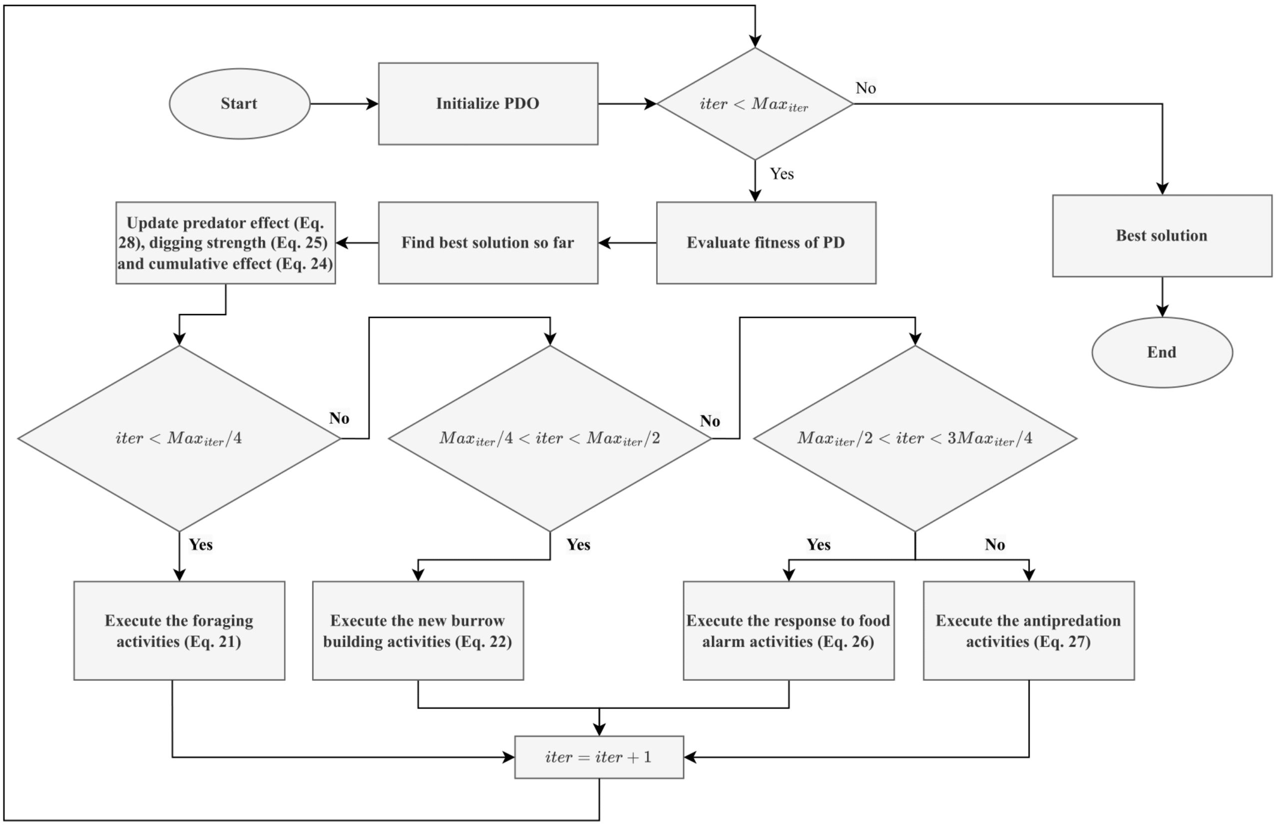

Figure 4. PDO flowchart [17]

3. EXPERIMENTAL DESIGN

In this section, the parameters of the metaheuristics and the test functions to be used in comparisons are given.

3.1. Parameters

Each metaheuristic has its own parameter and is given in Table 1. ARO does not require any algorithm-specific control parameters, as stated by Wang et al. [8]. Therefore, it is not listed in this table. p_1, p_2, p_3 are the parameters for selecting strategies in the exploration and exploitation. The parameters L represents the probabilities associated with the selection of the best vulture. w is the parameter whether the exploration and exploitation will be terminated. p_c gives the crossover probability and P_m represents the probability of mutation. ρ is the food source alarm parameter. ε is the food source quality parameter. Additionally, Δ is individual prairie dog difference.

3.2. Test functions

In this paper, 8 different test functions were used to compare metaheuristics. Among the functions presented in Table 2, F_1, F_2, F_3 and F_4 are unimodal, F_5, F_6, F_7 and F_8 are multimodal. In addition, F_{min} represents the optimum value and the range represents the boundaries of the search space of the functions. The dimension value for all functions is taken as 10, 30, 50 respectively. Furthermore, the plots of the functions are given in Figure 5.

Table 1. Parameters of metaheuristics

Metaheuristic	Parameter	Value
AVOA	p_1	0.6
	p_2	0.4
	p_3	0.6
	L_1	0.8
	L_2	0.2
	w	2.5
GA	p_c	0.80
	P_m	0.20
PDO	ρ	0.1
	ε	2.2204E-16
	Δ	0.005
All metaheuristics	Population size	50
	Max. iterations	1000

A total of 8 benchmark functions were selected to ensure a balanced and representative evaluation of the algorithms' capabilities. Four unimodal functions were included to assess the exploitation performance, i.e., the algorithms' ability to converge quickly to a single global optimum. Four multimodal functions were chosen to evaluate exploration performance, reflecting the ability to escape local optima and explore the solution space broadly. Selecting 4 functions from each category also helps maintain computational efficiency while allowing statistically meaningful analysis across different dimensions.

Table 2. Test functions

Function	Dimension	Range	F_{min}
$F_1(x) = \sum_{i=1}^{n} x_i^2$	10,30,50	[-100,100]	0
$F_2(x) = \sum_{i=0}^{n} \lvert x_i \rvert + \prod_{i=0}^{n} \lvert x_i \rvert$	10,30,50	[-10,10]	0
$F_3(x) = \sum_{i=1}^{n-1} [100(x_i - x_{i+1})^2 + (1 - x_i)^2]$	10,30,50	[-30,30]	0
$F_4(x) = \sum_{i=1}^{n-1} = \sum_{i=1}^{n} ([x_i - 0.5])^2$	10,30,50	[-100,100]	0
$F_5(x) = 10 + \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i)\right)$	10,30,50	[-5.12,5,12]	0
$F_6(x) = -\text{aexp}\left(-0.02\sqrt{n^{-1}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(n^{-1}\sum_{i=1}^{n} \cos(2\pi x_i)\right) + a + e, a = 20$	10,30,50	[-32,32]	0
$F_7(x) = 1 + \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right)$	10,30,50	[-600,600]	0
$F_8(x) = 0.1(\sin^2(3\pi x_1) + \sum_{i=1}^{n} (x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 1 + \sin^2(2\pi x_n)) + \sum_{i=1}^{n} u(x_i, 5,100,4)$	10,30,50	[-50,50]	0

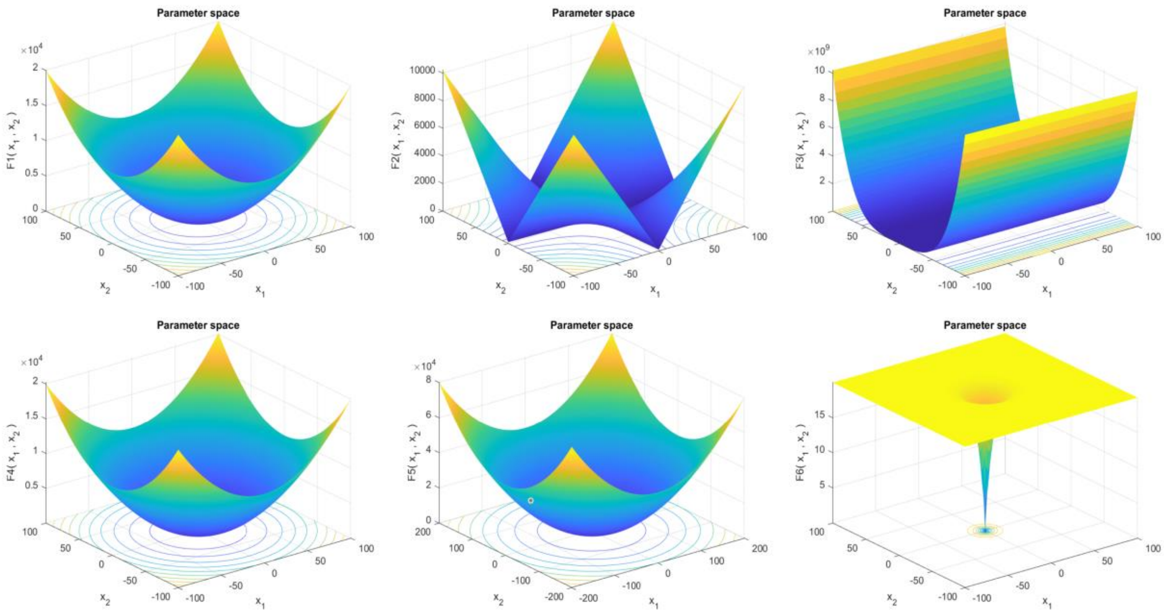

Figure 5. Test functions (F_1- F_8)

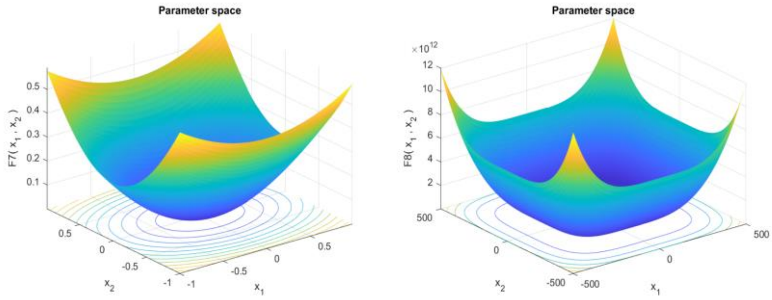

Figure 5. Test functions (F_1- F_8) (continued)

3.3. Engineering problems

In this study, three well-known engineering design problems are investigated: pressure vessel design, tension/compression spring design, and three-bar truss design. This section presents the mathematical formulation of each problem in detail.

3.3.1. Pressure Vessel Design

In this engineering design problem, the goal is to minimize the total cost associated with constructing a cylindrical pressure vessel. The design involves four decision variables: the shell thickness ($z_1 = T_s$), the head thickness ($z_2 = T_h$), the inner radius of the vessel ($z_3 = r$), and the length of the cylindrical section without the head ($z_4 = L$).

The objective function, which represents the cost, is subject to four nonlinear constraints related to structural and volume requirements. The mathematical representation of the problem is given below:

Design vector: $\vec{z} = [z_1, z_2, z_3, z_4] = [T_s, T_h, r, L]$
Objective function:

$$f(\vec{z}) = 1.7781 z_2 z_3^3 + 3.1661 z_1^2 z_4 + 19.84 z_1^2 z_3 + 0.6224 z_1 z_3 z_4$$

Constraints:

$$g_1(\vec{z}) = -z_1 + 0.0193 z_3 \leq 0$$

$$g_2(\vec{z}) = -z_3 + 0.00954 z_3 \leq 0$$

$$g_3(\vec{z}) = -\pi z_3^2 z_4 - \frac{4}{3}\pi z_3^3 + 1{,}296{,}000 \leq 0$$

$$g_4(\vec{z}) = z_4 - 240 \leq 0$$

Variable bounds:

$$0 \leq z_1, z_2 \leq 99, 10 \leq z_3, z_4 \leq 200$$

This formulation ensures that the structural integrity and volume constraints are met while minimizing material and manufacturing costs.

3.3.2. Tension/Compression Spring Design

This optimization problem focuses on minimizing the weight of a tension or compression spring. The design involves three key variables: the wire diameter ($z_1 = d$), the mean coil diameter ($z_2 = D$), and the number of active coils ($z_3 = N$). These parameters determine the physical structure of the spring.

The objective is to reduce the spring's weight while ensuring it satisfies several mechanical constraints, including limits on deflection, shear stress, and surge frequency. The mathematical formulation of the problem is as follows:

Design vector: $\vec{z} = [z_1, z_2, z_3] = [d, D, N]$
Objective function:

$$f(\vec{z}) = (z_3 + 2) z_2 z_1^2$$

Constraints:

$$g_1(\vec{z}) = 1 - \frac{z_2^3 z_3}{71785 z_1^4} \leq 0$$

$$g_2(\vec{z}) = \frac{1}{5108 z_1^2} + \frac{4z_2^2 - z_1 z_2}{12566(z_2 z_1^3 - z_1^4)} \leq 0$$

$$g_3(\vec{z}) = 1 - \frac{140.45 z_1}{z_2^2 z_3} \leq 0$$

$$g_4(\vec{z}) = \frac{z_1 + z_2}{1.5} - 1 \leq 0$$

This formulation ensures that the spring design is both lightweight and structurally feasible under mechanical and dynamic load conditions.

3.3.3. The Three-Bar Truss Design Problem

This structural optimization problem aims to minimize the total weight of a simple planar truss system subjected to external loading. The design involves two decision variables: the cross-sectional areas of two different truss elements, denoted as $x_1 = A_1$ and $x_2 = A_2$. These parameters directly affect the truss's weight and its mechanical behavior under stress. The objective function is defined as a function of the material length and the cross-sectional areas, while the design is subject to multiple nonlinear constraints, including limits on deflection, buckling, and maximum allowable stress. These constraints ensure the structural integrity of the truss under given loading conditions. The mathematical formulation is expressed as follows:

Design vector: $\vec{X} = [x_1, x_2] = [A_1, A_2]$
Objective function:

$$f(\vec{X}) = \left(2\sqrt{2}x_1 + x_2\right) \times L$$

Constraints:

$$g_1(\vec{X}) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1 x_2} P - \sigma \leq 0$$

$$g_2(\vec{X}) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1 x_2} P - \sigma \leq 0$$

$$g_3(\vec{X}) = \frac{1}{\sqrt{2}x_2 + x_1} P - \sigma \leq 0$$

Variable bounds: $0 < x_1, x_2 \leq 1$

where: $L = 100$ cm (length of each truss member), $P = 2\text{kN/cm}^2$ (applied load), $\sigma = 2\text{kN/cm}^2$ (maximum allowable stress).

This problem is widely used in the literature to test the effectiveness of optimization algorithms under multi-constraint structural design scenarios, as it combines weight minimization with critical mechanical limitations.

4. Result and Discussion

In order to provide a more comprehensive and structured evaluation of the metaheuristic algorithms, this section is divided into four subsections. First, the simulation results of the test functions are analyzed in terms of mean, best, and worst values. Then, the convergence behaviors of the algorithms are investigated by plotting convergence curves for different dimensions. In the third subsection, a statistical evaluation is conducted using a one-tailed t-test to determine whether the observed performance differences are significant. Finally, a new subsection is introduced to analyze the computational time and complexity of the algorithms. This addition directly addresses reviewer comments regarding the importance of computational cost in real-world applications.

4.1. Benchmark Function Results

In this paper, metaheuristics were run independently 30 times. MATLAB R2022b platform was used for performance analysis of metaheuristics. Simulations were implemented on a machine with AMD Ryzen 5 3500X CPU, 3.6 GHz speed and 16 GB RAM. Simulation results of the test functions are given in Tables 3, 4 and 5, respectively. The mean, standard deviation, best and worst results of all functions are presented in these tables. Additionally, the metaheuristics with the lowest mean and best value are bolded.

Table 3. Simulation results of test functions for dim = 10

Functions	Algorithms	ARO	AVOA	GA	PDO
F_1	Mean	2.99E-141	**0.00E+00**	2.04E+03	**0.00E+00**
	Std	1.44E-140	0.00E+00	5.17E+02	0.00E+00
	Best	2.56E-152	**0.00E+00**	8.01E+02	**0.00E+00**
	Worst	8.03E-140	0.00E+00	2.83E+03	0.00E+00
F_2	Mean	5.71E-74	**0.00E+00**	1.07E+01	**0.00E+00**
	Std	2.41E-73	0.00E+00	1.85E+00	0.00E+00
	Best	3.36E-82	**0.00E+00**	6.44E+00	**0.00E+00**
	Worst	1.33E-72	0.00E+00	1.38E+01	0.00E+00
F_3	Mean	1.36E+00	**4.81E-06**	1.78E+04	4.28E+00
	Std	1.80E+00	6.22E-06	3.76E+03	3.65E+00
	Best	5.94E-06	**4.81E-10**	1.03E+04	6.35E-02
	Worst	4.34E+00	2.03E-05	2.57E+04	9.00E+00
F_4	Mean	1.71E-20	1.53E-16	2.24E+03	**2.97E-30**
	Std	3.60E-20	2.02E-16	5.87E+02	1.18E-29
	Best	4.33E-24	5.01E-18	9.15E+02	**3.08E-33**
	Worst	1.8E-19	8.67E-16	3.20E+03	6.43E-29
F_5	Mean	**0.00E+00**	**0.00E+00**	4.99E+02	**0.00E+00**
	Std	0.00E+00	0.00E+00	6.29E+00	0.00E+00
	Best	**0.00E+00**	**0.00E+00**	4.80E+02	**0.00E+00**
	Worst	0.00E+00	0.00E+00	5.08E+02	0.00E+00
F_6	Mean	**4.44E-16**	**4.44E-16**	1.35E+01	**4.44E-16**
	Std	0.00E+00	0.00E+00	1.27E+00	0.00E+00
	Best	**4.44E-16**	**4.44E-16**	1.05E+01	**4.44E-16**
	Worst	4.44E-16	4.44E-16	1.53E+01	4.44E-16
F_7	Mean	**0.00E+00**	**0.00E+00**	2.09E+01	**0.00E+00**
	Std	0.00E+00	0.00E+00	5.58E+00	0.00E+00
	Best	**0.00E+00**	**0.00E+00**	1.25E+01	**0.00E+00**
	Worst	0.00E+00	0.00E+00	3.07E+01	0.00E+00
F_8	Mean	**2.43E-18**	1.83E-14	1.12E+06	7.48E-01
	Std	1.22E-17	2.53E-14	7.49E+05	3.70E-01
	Best	**8.52E-25**	5.77E-16	1.07E+05	1.97E-02
	Worst	6.82E-17	1.16E-13	2.85E+06	1.00E+00

When all simulation results are evaluated, AVOA and PDO reach the optimum value in F_1, F_2, F_5 and F_7, while ARO reaches the optimum value in F_5 ve F_7. ARO follows AVOA and PDO with very small differences in F_1 ve F_2 where it does not reach an optimum value. GA, on the other hand, did not achieve the optimum value in any function and was the worst performing metaheuristic. In F_6 function, the newly proposed metaheuristics performed the same. In the simulation results of dimension 10, AVOA has the best and lowest mean values in F_3, PDO in F_4 and ARO in F_8. In the other dimensions, AVOA performs better in these functions. In dimension 10, the new metaheuristics have the largest difference in F_8, with an approximate value of 1.97E-02. For dimension 30, the biggest difference in the best values that the GA has is in F_8 and is about 7.37E+07. In dimension 50, the GA has the largest difference of 1.14E+11 (F_2) and 6.53E+08 (F_8) in the means and best values for all functions.

Table 4. Simulation results of test functions for dim = 30

Functions	Algorithms	ARO	AVOA	GA	PDO
F_1	Mean	9.07E-124	**0.00E+00**	3.24E+04	**0.00E+00**
	Std	4.96E-123	0.00E+00	2.85E+03	0.00E+00
	Best	1.05E-143	**0.00E+00**	2.64E+04	**0.00E+00**
	Worst	2.72E-122	0.00E+00	3.69E+04	0.00E+00
F_2	Mean	2.74E-70	**0.00E+00**	1.01E+03	**0.00E+00**
	Std	6.50E-70	0.00E+00	1.66E+03	0.00E+00
	Best	6.75E-76	**0.00E+00**	8.76E+01	**0.00E+00**
	Worst	3.04E-69	0.00E+00	6.34E+03	0.00E+00
F_3	Mean	1.67E+00	**1.83E-06**	3.63E+05	6.80E+00
	Std	6.33E+00	1.98E-06	3.81E+04	1.05E+01
	Best	1.75E-04	**4.94E-08**	2.91E+05	3.24E-01
	Worst	2.52E+01	7.63E-06	4.39E+05	2.90E+01
F_4	Mean	1.86E-06	**1.64E-10**	3.36E+04	3.46E+00
	Std	1.63E-06	1.21E-10	2.68E+03	2.08E+00
	Best	4.46E-08	**2.85E-11**	2.71E+04	1.60E-01
	Worst	5.60E-06	5.35E-10	3.95E+04	7.25E+00
F_5	Mean	**0.00E+00**	**0.00E+00**	4.65E+03	**0.00E+00**
	Std	0.00E+00	0.00E+00	1.42E+01	0.00E+00
	Best	**0.00E+00**	**0.00E+00**	4.62E+03	**0.00E+00**
	Worst	0.00E+00	0.00E+00	4.68E+03	0.00E+00
F_6	Mean	**4.44E-16**	**4.44E-16**	1.93E+01	**4.44E-16**
	Std	0.00E+00	0.00E+00	2.07E-01	0.00E+00
	Best	**4.44E-16**	**4.44E-16**	1.89E+01	**4.44E-16**
	Worst	4.44E-16	4.44E-16	1.96E+01	4.44E-16
F_7	Mean	**0.00E+00**	**0.00E+00**	3.00E+02	**0.00E+00**
	Std	0.00E+00	0.00E+00	2.58E+01	0.00E+00
	Best	**0.00E+00**	**0.00E+00**	2.44E+02	**0.00E+00**
	Worst	0.00E+00	0.00E+00	3.40E+02	0.00E+00
F_8	Mean	7.39E-04	**4.80E-10**	2.38E+08	2.93E+00
	Std	2.79E-03	3.05E-10	5.42E+07	1.44E-01
	Best	5.88E-08	**1.47E-10**	7.37E+07	2.44E+00
	Worst	1.10E-02	1.39E-09	3.23E+08	3.00E+00

Table 5. Simulation results of test functions for dim = 50

Functions	Algorithms	ARO	AVOA	GA	PDO
F_1	Mean	8.80E-122	**0.00E+00**	7.70E+04	**0.00E+00**
	Std	3.21E-121	0.00E+00	4.59E+03	0.00E+00
	Best	2.21E-137	**0.00E+00**	6.71E+04	**0.00E+00**
	Worst	1.56E-120	0.00E+00	8.48E+04	0.00E+00
F_2	Mean	2.10E-66	**0.00E+00**	1.14E+11	**0.00E+00**
	Std	8.24E-66	0.00E+00	3.03E+11	0.00E+00
	Best	8.95E-75	**0.00E+00**	2.11E+06	**0.00E+00**
	Worst	4.14E-65	0.00E+00	1.40E+12	0.00E+00
F_3	Mean	2.18E-02	**4.28E-06**	9.71E+05	1.81E+01
	Std	2.05E-02	3.63E-06	6.27E+04	2.01E+01
	Best	3.19E-03	**2.06E-07**	8.13E+05	1.14E+00
	Worst	9.02E-02	1.85E-05	1.08E+06	4.90E+01
F_4	Mean	2.28E-04	**1.83E-08**	7.75E+04	7.52E+00
	Std	1.03E-04	1.26E-08	3.44E+03	2.52E+00
	Best	7.22E-05	**4.59E-09**	6.91E+04	2.83E+00
	Worst	4.12E-04	6.22E-08	8.30E+04	1.23E+01
F_5	Mean	**0.00E+00**	**0.00E+00**	1.28E+04	**0.00E+00**
	Std	0.00E+00	0.00E+00	1.54E+01	0.00E+00
	Best	**0.00E+00**	**0.00E+00**	1.28E+04	**0.00E+00**
	Worst	0.00E+00	0.00E+00	1.29E+04	0.00E+00
F_6	Mean	**4.44E-16**	**4.44E-16**	2.00E+01	**4.44E-16**
	Std	0.00E+00	0.00E+00	9.99E-02	0.00E+00
	Best	**4.44E-16**	**4.44E-16**	1.98E+01	**4.44E-16**
	Worst	4.44E-16	4.44E-16	2.02E+01	4.44E-16
F_7	Mean	**0.00E+00**	**0.00E+00**	6.69E+02	**0.00E+00**
	Std	0.00E+00	0.00E+00	5.36E+01	0.00E+00
	Best	**0.00E+00**	**0.00E+00**	5.25E+02	**0.00E+00**
	Worst	0.00E+00	0.00E+00	7.43E+02	0.00E+00
F_8	Mean	6.28E-03	**2.11E-09**	8.65E+08	4.85E+00
	Std	1.93E-02	1.98E-09	9.42E+07	7.08E-01
	Best	1.32E-05	**1.07E-10**	6.53E+08	1.13E+00
	Worst	9.74E-02	8.32E-09	1.00E+09	5.00E+00

Table 6. Metaheuristics total rank

Metaheuristic	Best	Mean
ARO	10	10
AVOA	**22**	**22**
GA	0	0
PDO	16	16

The numbers of providing the best and mean result in terms of metaheuristics in all simulation results are given in Table 6. According to the table, in terms of overall success, the best metaheuristic is AVOA, followed by PDO, ARO, and GA, respectively. The most successful values are bolded in the table. AVOA achieved 6 more best results than the closest PDO. Among the new proposed ones, ARO seems to be the most unsuccessful, but it produced very close values compared to the others.

4.2. Convergence curves of functions

The convergence curve is a graph that shows how quickly or accurately a mathematical or computational process progresses towards a specific goal or outcome [19]. The convergence plots of metaheuristics on test functions for dimensions 10, 30, and 50 are provided in Figures 6-7.

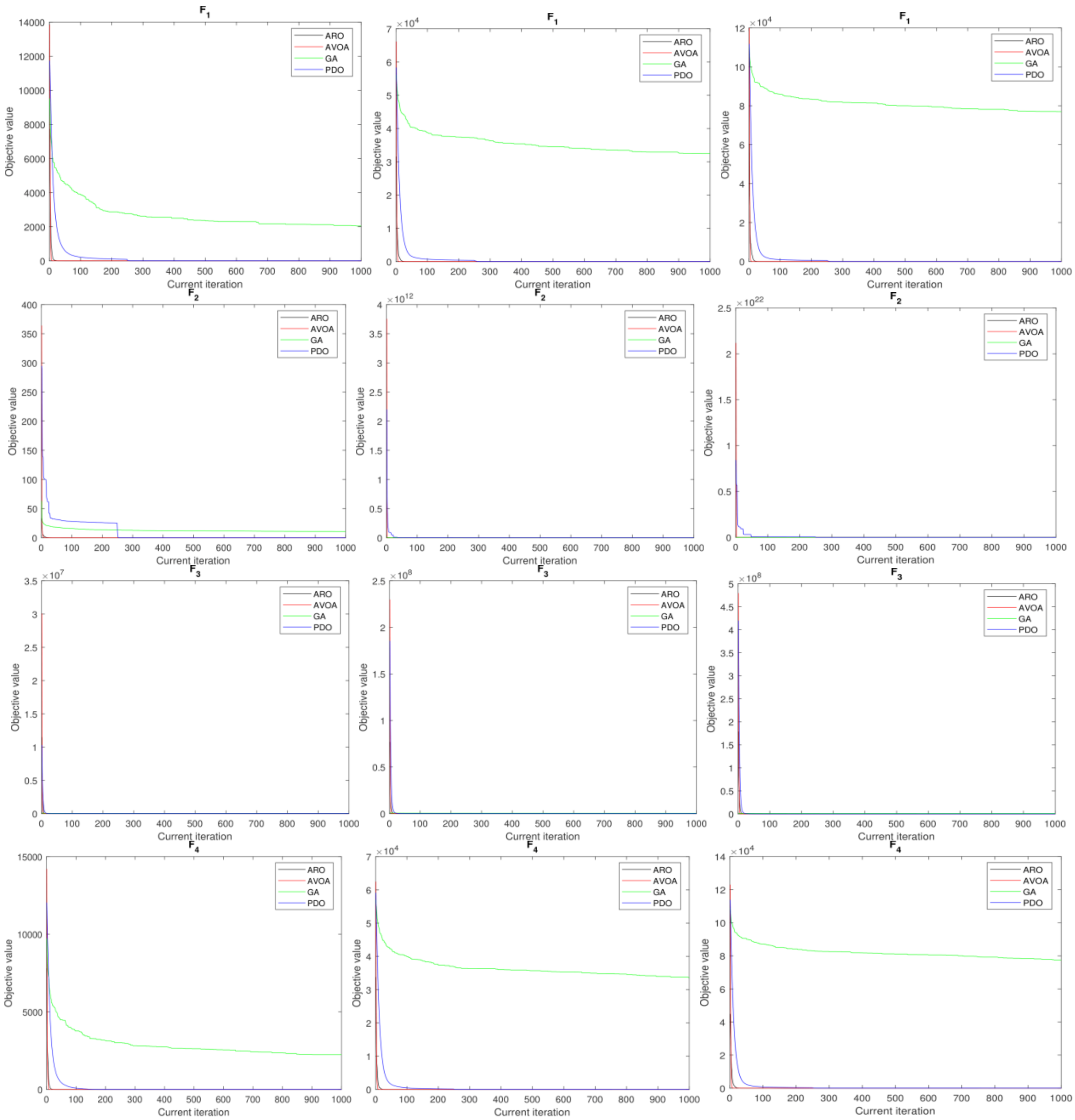

Figure 6. Convergence curves of test functions (F_1 - F_4)

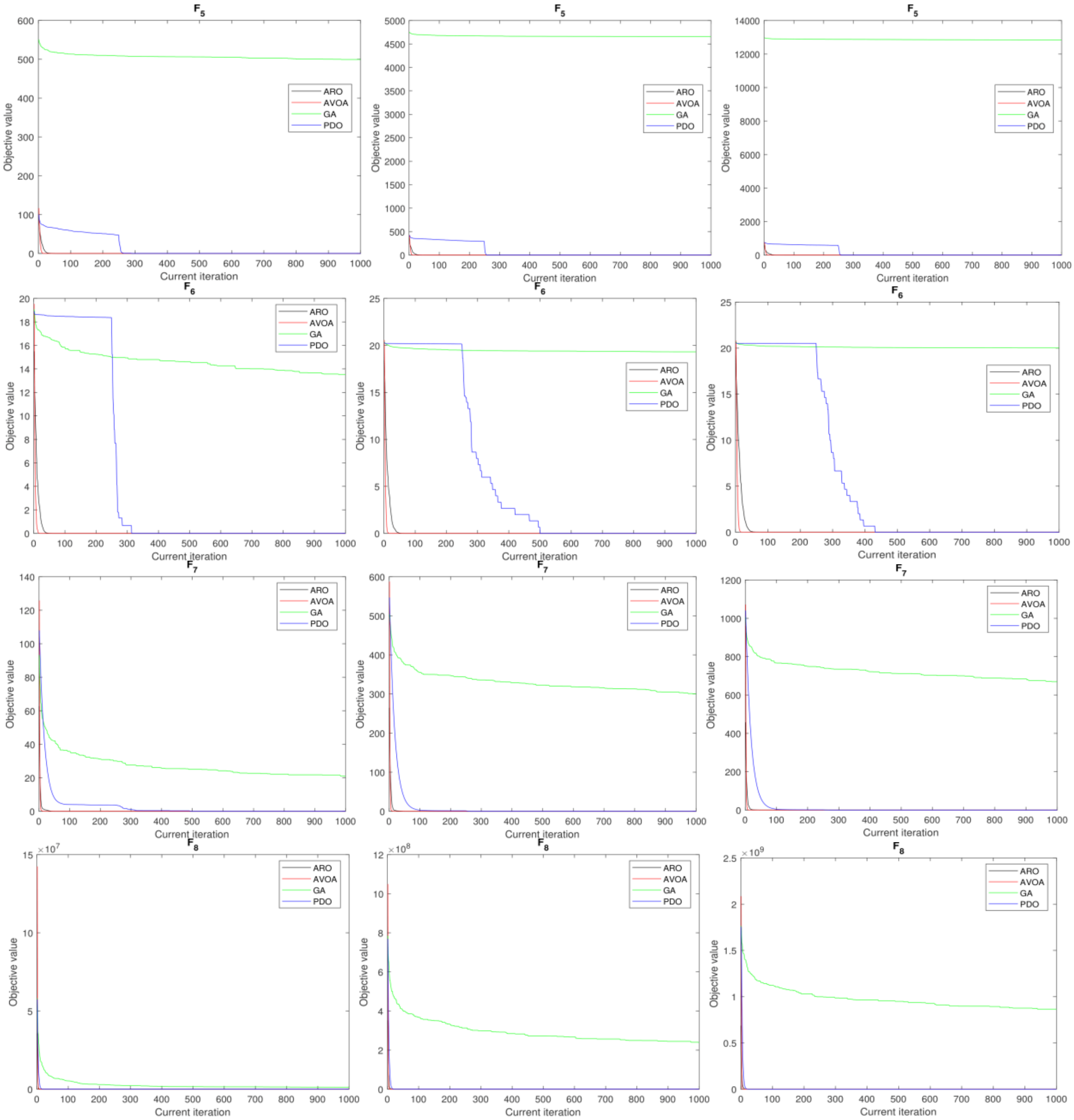

Figure 7. Convergence curves of test functions (F_5 - F_8)

AVOA showed faster convergence for each dimension in all functions. AVOA is followed by ARO. Even though PDO shows slower convergence than ARO, it has better performance in simulation results as seen in Table 6. AVOA and PDO produced the same values after a certain iteration for all dimensions in F_1, F_2, F_5, F_6, and F_7. Similarly, ARO terminated by producing the same values after a certain iteration for all dimensions in F_5, F_6, and F_7. Additionally, the convergence speed of the metaheuristics became slower as the dimension increased. In contrast to the newly proposed metaheuristics, GA performed worse and showed less convergence.

4.3. t-test result

The t-test is a statistical method used to determine whether there is a significant difference between the means of two groups [20, 21]. In this study, a one-tailed t-test was performed on the results of the metaheuristic simulations with a 5% significance level ($\alpha = 0.05$). A result where $h = 1$ and $p < 0.05$ indicates that the metaheuristic algorithm on the right side of the pairwise comparison outperformed the other with statistical significance.

The detailed p-values and h-values for each function and dimension are provided in Tables 7, 8, and 9. However, it is also crucial to interpret the practical implications of these statistical differences in terms of algorithm performance.

Table 7. t-test results (dim=10)

Functions	GA-ARO		GA-AVOA		GA-PDO		ARO-AVOA		ARO-PDO		PDO-AVOA	
	p	*h*	*p*	*h*	*p*	*h*	*p*	*h*	*p*	*h*	*p*	*h*
F_1	9.29E-20	**1**	9.29E-20	**1**	9.29E-20	**1**	1.40E-01	0	1.40E-01	0	-	-
F_2	2.13E-24	**1**	2.13E-24	**1**	2.13E-24	**1**	1.10E-01	0	1.10E-01	0	-	-
F_3	6.03E-22	**1**	6.02E-22	**1**	6.01E-22	**1**	1.64E-04	**1**	1.00E+00	**1**	2.58E-07	**1**
F_4	2.42E-19	**1**	2.42E-19	**1**	2.42E-19	**1**	1.00E+00	0	7.90E-03	**1**	1.00E+00	0
F_5	3.65E-57	**1**	3.65E-57	**1**	3.65E-57	**1**	-	-	-	-	-	-
F_6	5.92E-32	**1**	5.92E-32	**1**	5.92E-32	**1**	-	-	-	-	-	-
F_7	4.06E-19	**1**	4.06E-19	**1**	4.06E-19	**1**	-	-	-	-	-	-
F_8	2.33E-09	**1**	2.33E-09	**1**	2.33E-09	**1**	1.00E+00	0	1.00E+00	0	3.12E-12	**1**

Table 8. t-test results (dim=30)

Functions	GA-ARO		GA-AVOA		GA-PDO		ARO-AVOA		ARO-PDO		PDO-AVOA	
	p	*h*	*p*	*h*	*p*	*h*	*p*	*h*	*p*	*h*	*p*	*h*
F_1	1.00E-32	**1**	1.00E-32	**1**	1.00E-32	**1**	1.60E-01	0	1.60E-01	0	-	-
F_2	1.20E-03	**1**	1.20E-03	**1**	1.20E-03	**1**	1.00E-02	**1**	1.00E-02	**1**	-	-
F_3	1.63E-30	**1**	1.63E-30	**1**	1.63E-30	**1**	7.97E-02	0	9.99E-01	0	6.51E-04	**1**
F_4	5.65E-34	**1**	5.65E-34	**1**	5.65E-34	**1**	4.06E-07	**1**	1	0	2.46E-10	**1**
F_5	2.17E-74	**1**	2.17E-74	**1**	2.17E-74	**1**	-	-	-	-	-	-
F_6	9.44E-59	**1**	9.44E-59	**1**	9.44E-59	**1**	-	-	-	-	-	-
F_7	4.83E-33	**1**	4.83E-33	**1**	4.83E-33	**1**	-	-	-	-	-	-
F_8	5.24E-21	**1**	5.24E-21	**1**	5.24E-21	**1**	7.85E-02	0	1	0	4.70E-40	**1**

Table 9. t-test results (dim=50)

Functions	GA-ARO		GA-AVOA		GA-PDO		ARO-AVOA		ARO-PDO		PDO-AVOA	
	p	*h*	*p*	*h*	*p*	*h*	*p*	*h*	*p*	*h*	*p*	*h*
F_1	1.31E-37	**1**	1.31E-37	**1**	1.31E-37	**1**	7.18E-02	0	7.18E-02	0	-	-
F_2	2.42E-02	**1**	2.42E-02	**1**	2.42E-02	**1**	9.00E-02	0	9.00E-02	0	-	-
F_3	1.80E-36	**1**	1.80E-36	**1**	1.80E-36	**1**	1.31E-06	**1**	1	0	1.55E-05	**1**
F_4	2.55E-41	**1**	2.55E-41	**1**	2.56E-41	**1**	3.85E-13	**1**	1	0	2.00E-16	**1**
F_5	3.36E-72	**1**	3.36E-72	**1**	3.36E-72	**1**	-	-	-	-	-	-
F_6	6.55E-68	**1**	6.55E-68	**1**	6.55E-68	**1**	-	-	-	-	-	-
F_7	6.50E-34	**1**	6.50E-34	**1**	6.50E-34	**1**	-	-	-	-	-	-
F_8	4.53E-30	**1**	4.53E-30	**1**	4.53E-30	**1**	4.22E-02	**1**	1	0	1.91E-26	**1**

According to Tables 7,8 and 9, the evaluations are as follows:

- In all dimensions (10, 30, 50), GA performed significantly worse than the newly proposed metaheuristics (ARO, AVOA, PDO), both statistically and practically. The mean values for GA were multiple orders of magnitude higher, showing weak convergence and poor optimization capacity across all test functions.
- While AVOA consistently outperformed ARO and PDO in most cases, especially in multimodal functions, the differences between AVOA and the other new algorithms were generally small in practical terms, even when statistically significant (e.g., differences on the order of 10^{-4} to 10^{-6}).
- PDO showed strong robustness, occasionally outperforming ARO in certain functions, such as those involving higher dimensions and complex landscapes. Although t-tests confirmed some of these differences as statistically significant, in real-world usage, the differences would yield very similar results from a practical optimization perspective.
- The statistically significant superiority of AVOA in several functions (e.g., F_3, F_4, F_8) highlights its strong exploration and exploitation balance, leading to better convergence. However, in functions where no significant difference was observed between AVOA and PDO or ARO, it is inferred that all three algorithms can be safely considered high-performing alternatives for benchmark problems.

In conclusion, while t-tests reveal which differences are statistically significant, the practical value lies in the consistency, speed of convergence, and robustness of the algorithms. AVOA leads in most metrics, but PDO and ARO offer competitive results, especially when computational cost or algorithm simplicity is considered.

4.4. Engineering problem results

The optimization results for the engineering design problems are evaluated across three benchmark cases: Pressure Vessel Design, Tension/Compression Spring Design, and Three-Bar Truss Design. The best run results obtained for each problem, along with the corresponding parameter values, are presented in Tables 10–12.

Table 10. Best results for the pressure vessel design problem

Algorithm	T_s	T_h	r	L	Optimal value
ARO	0.7782	0.3847	40.3217	199.9712	5.8854e+03
AVOA	0.7790	0.3851	40.3639	10.0010	5.8868e+03
GA	1.2918	0.5706	42.2641	12.7864	1.1017e+04
PDO	0.8156	0.3975	40.8373	10.0000	6.0308e+03

According to Table 10, ARO achieved the best result with the lowest cost (5.8854e+03), followed closely by AVOA (5.8868e+03). GA performed the worst with a significantly higher cost (1.1017e+04), while PDO showed moderate performance (6.0308e+03). The longer cylinder length in ARO's design may have contributed to its better outcome.

Table 11. Best results for the tension/compression spring design

Algorithm	d	D	N	Optimal value
ARO	0.0516	0.3537	10.4114	0.0127
AVOA	0.0500	0.3174	4.2926	0.0127
GA	0.0534	0.3572	2.2595	0.0139
PDO	0.0500	0.3116	8.1058	0.0127

According to Table 11, ARO, AVOA, and PDO all achieved the same optimal value (0.0127), indicating equal performance. GA performed worse with a higher cost (0.0139). Although the optimal values are the same for the three algorithms, the design parameters differ, especially in the number of active coils (N).

Table 12. Best results for the three bar truss design

Algorithm	A_1	A_2	Optimal value
ARO	0.7887	0.4082	263.8958
AVOA	0.7835	0.3640	263.8958
GA	0.7624	0.2454	263.9174
PDO	0.7356	0.4074	263.8958

According to Table 12, ARO, AVOA, and PDO achieved the same optimal value (263.8958), indicating equally successful performance. GA resulted in a slightly higher value (263.9174), making it the least effective among the four. Despite identical outcomes for three algorithms, their parameter values differ slightly, showing that multiple configurations can lead to the same objective value. To provide further insight into the optimization behavior, convergence curves for all three engineering design problems are presented in Figure 8.

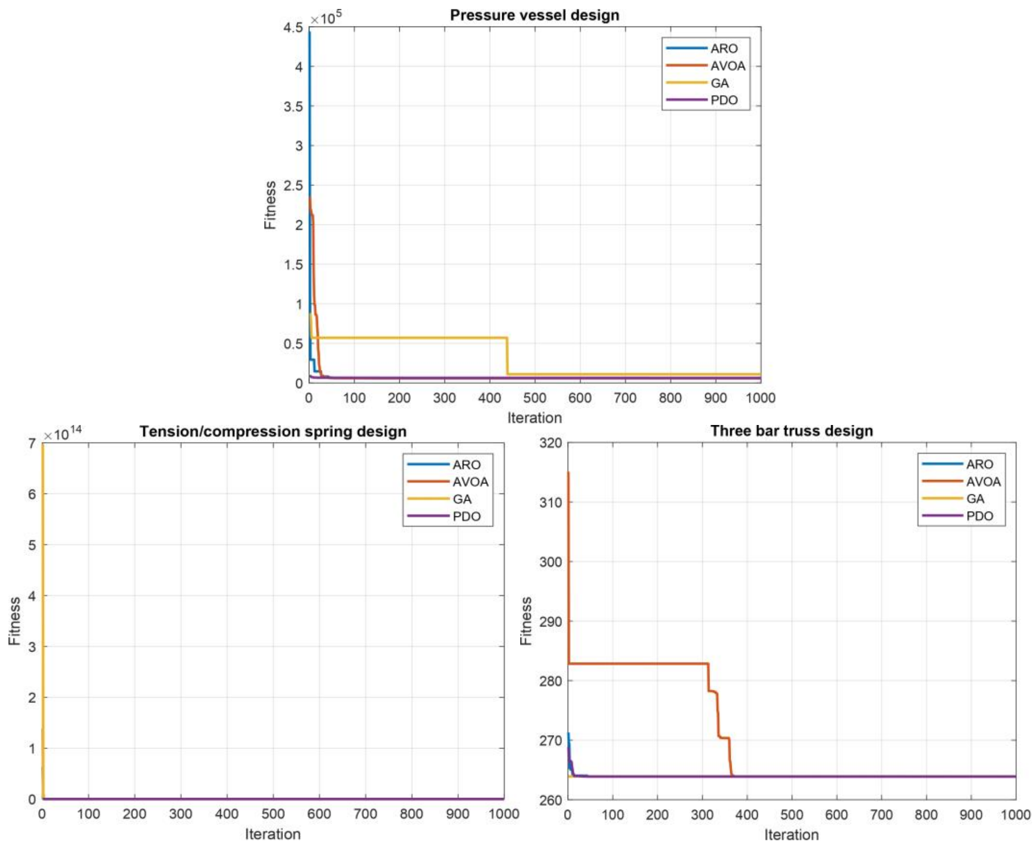

Figure 8. Convergence curves of the algorithms on three engineering problems

The convergence plots reveal that ARO generally demonstrates the fastest and most stable convergence across the problems. In the spring design, ARO, AVOA, and PDO exhibit nearly identical and immediate convergence behavior, indicating comparable effectiveness. In the truss design, ARO and PDO reach the optimum quickly, while AVOA converges more slowly and GA performs moderately. For the pressure vessel problem, ARO achieves the fastest convergence, closely followed by PDO, both clearly outperforming AVOA and GA.

4.4. Computational Complexity Analysis

In addition to the performance outcomes, the algorithms were analyzed in terms of their computational characteristics. While all algorithms were run under identical conditions (population size = 50, iterations = 1000), they differ in the internal number of operations per iteration. Table 13 summarizes the estimated per-iteration computational complexity for each algorithm.

Table 13. Approximate computational complexity per iteration

Algorithm	Complexity	Structural Notes
GA	$O(n)$	Basic evolutionary operators
PDO	$O(n)$	Simple interaction-based updates
AVOA	$O(n \times d)$	Phase switching and adaptive terms
ARO	$O(n \times d + n \log n)$	Behavioral modeling and sorting operations

$n = population\ size, d = problem\ dimension.$

It can be inferred from Table 10 that GA and PDO involve relatively simple operations, while AVOA and ARO include more complex mechanisms such as phase switching, adaptive behaviors, and sorting processes. These internal differences may influence total processing time, especially in high-dimensional problems, although all algorithms remain feasible for standard benchmark testing.

5. CONCLUSION

In this paper, the performance analysis of ARO, AVOA, PDO and GA metaheuristics is conducted for the first time using 8 different test functions. The analyses revealed that the newly proposed metaheuristics outperformed the popular GA. AVOA was observed to be the most successful, producing better results as referenced in Table 6. Other newly proposed is performed slightly worse than AVOA. The obtained simulation results were evaluated for statistical significance using t-tests. The tests indicated that there were minimal significant differences among the newly proposed methods, but these differences were more expressed when compared to the popular GA. In future studies, the performance of ARO, AVOA, and PDO will be compared with other modern metaheuristics such as Particle Swarm Optimization (PSO), Differential Evolution (DE), and Artificial Bee Colony (ABC) under similar conditions to expand the scope of the evaluation. It is also aimed to solve different optimization problems with these metaheuristics.

REFERENCES

[1] Yang XS. Nature-inspired metaheuristic algorithms. 2nd ed. Frome (UK): Luniver Press; 2008.
[2] Jia H, Rao H, Wen C, Mirjalili S. Crayfish optimization algorithm. Artif Intell Rev. 2023;1–61.
[3] Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks. Vol. 4. IEEE; 1995. p. 1942–8.
[4] Mirjalili S. Genetic algorithm. In: Mirjalili S, editor. Evolutionary algorithms and neural networks: theory and applications. Cham: Springer; 2019. p. 43–55.
[5] Karaboga D. Artificial bee colony algorithm. Scholarpedia. 2010;5(3):6915.
[6] Dorigo M, Birattari M, Stützle T. Ant colony optimization. IEEE Comput Intell Mag. 2006;1(4):28–39.
[7] Dehghani M, Montazeri Z, Trojovská E, Trojovský P. Coati optimization algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems. Knowl Based Syst. 2023;259:110011.
[8] Wang L, Cao Q, Zhang Z, Mirjalili S, Zhao W. Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems. Eng Appl Artif Intell. 2022;114:105082.
[9] Tynes VV, editor. Behavior of exotic pets. Hoboken: John Wiley & Sons; 2010.
[10] Camp MJ, Rachlow JL, Shipley LA, Johnson TR, Bockting KD. Grazing in sagebrush rangelands in western North America: implications for habitat quality for a sagebrush specialist, the pygmy rabbit. Rangel J. 2014;36(2):151–9.
[11] Abdollahzadeh B, Gharehchopogh FS, Mirjalili S. African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems. Comput Ind Eng. 2021;158:107408.
[12] Xue Y, Jia W, Zhao X, Pang W. An evolutionary computation based feature selection method for intrusion detection. Secur Commun Netw. 2018;2018:1–15.
[13] Yang XS, Deb S. Cuckoo search via Lévy flights. In: Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC). IEEE; 2009. p. 210–4.
[14] Yang XS. A new metaheuristic bat-inspired algorithm. In: González JR, Pelta DA, Cruz C, Terrazas G, Krasnogor N, editors. Nature inspired cooperative strategies for optimization (NICSO 2010). Berlin: Springer; 2010. p. 65–74.
[15] Gwiazda TD. Crossover for single-objective numerical optimization problems. Vol. 1. Wroclaw: Wroclaw University of Technology Press; 2006.
[16] Vasconcelos JA, Ramirez JA, Takahashi RHC, Saldanha RR. Improvements in genetic algorithms. IEEE Trans Magn. 2001;37(5):3414–7.
[17] Ezugwu AE, Agushaka JO, Abualigah L, Mirjalili S, Gandomi AH. Prairie dog optimization algorithm. Neural Comput Appl. 2022;34(22):20017–65.

[18] Hoogland JL. The black-tailed prairie dog: social life of a burrowing mammal. Chicago: University of Chicago Press; 1995.
[19] Yiğit H, Ürgün S, Mirjalili S. Comparison of recent metaheuristic optimization algorithms to solve the SHE optimization problem in MLI. Neural Comput Appl. 2023;35(10):7369–88.
[20] Kim TK. T test as a parametric statistic. Korean J Anesthesiol. 2015;68(6):540–6.
[21] Mishra P, Singh U, Pandey CM, Mishra P, Pandey G. Application of student's t-test, analysis of variance, and covariance. Ann Card Anaesth. 2019;22(4):407–11.