Uludağ University Journal of The Faculty of Engineering, Vol. 31, No. 1, 2026 RESEARCH

DOI: 10.17482/uumfd.1801249

BENCHMARKING SUCCESS: HOW MODERN METAHEURISTICS SOLVE COMPLEX ENGINEERING PROBLEMS

Metin KALYON [*]
Sibel ARSLAN [**]

Received: 10.10.2025; revised: 18.02.2026; accepted: 08.03.2026

Abstract: Recent developments in metaheuristic optimization algorithms have yielded significant and noteworthy results. These metaheuristics can additionally be utilized to evaluate engineering design challenges. In this study, 5 metaheuristics developed in recent years (Artificial Rabbit Optimization-ARO, Black Widow Optimization-BWO, Prairie Dog Optimization-PDO, Mountain Gazelle Optimization-MGO and Crayfish Optimization Algorithm -COA) success in engineering design problems was compared. To the best of our knowledge, this work represents the first comprehensive evaluation of these five metaheuristic algorithms on six well-known engineering design optimization problems: Tension/Compression Spring, Pressure Vessel, Welded Beam, Speed Reducer, Gear Set, and Three-Bar Truss. Upon assessing the experimental outcomes and convergence speeds, it becomes evident that the metaheuristic techniques employed in this research demonstrate effective efficacy against the challenges presented. Based on the obtained results, ARO achieved the highest performance, followed sequentially by BWO, MGO, COA, and PDO. In upcoming research, the goal is to employ additional metaheuristic techniques, particularly ARO, to address various engineering challenges.

Keywords: Metaheuristic Algorithms, Engineering Design Problems, Artificial Rabbit Optimization, Black Widow Optimization

Başarıyı Kıyaslama: Modern Metasezgiseller Karmaşık Mühendislik Problemlerini Nasıl Çözüyor?

Öz: Metaheuristik optimizasyon algoritmalarındaki son gelişmeler, önemli ve dikkate değer sonuçlar doğurmuştur.Bu metasezgisel yöntemler, ayrıca mühendislik tasarım zorluklarını değerlendirmek için de kullanılabilir. Bu çalışmada son yıllarda geliştirilen 5 metasezgiselin (Yapay Tavşan Optimizasyonu-ARO, Kara Dul Optimizasyonu-BWO, Çayır Köpeği Optimizasyonu-PDO, Dağ Ceylanı Optimizasyonu-MGO ve Kerevit Optimizasyon Algoritması-COA) mühendislik tasarım problemlerindeki başarıları karşılaştırılmıştır. Bildiğimiz kadarıyla, bu çalışma, gerilim/basınç yayı, basınçlı kap, kaynaklı kiriş, hız düşürücü, dişli seti ve üç çubuklu kafes gibi altı tanınmış mühendislik tasarım optimizasyon problemi üzerinde bu beş metaheuristik algoritmanın ilk kapsamlı değerlendirmesini temsil etmektedir. Deneysel sonuçlar ve yakınsama hızları değerlendirildiğinde, bu araştırmada kullanılan metasezgisel tekniklerin sunulan zorluklara karşı etkili bir etkinlik gösterdiği ortaya çıkmaktadır. Elde edilen sonuçlara göre en başarılı algoritma ARO olurken onu sırasıyla BWO, MGO, COA ve PDO takip etmektedir. Gelecekteki araştırmalarda, çeşitli mühendislik zorluklarını ele almak için özellikle ARO olmak üzere farklı metasezgisel tekniklerin kullanılması hedeflenmektedir.

Anahtar Kelimeler: Güncel Metasezgisel Algoritmalar, Mühendislik Tasarım Problemleri, Yapay Tavşan Optimizasyonu, Kara Dul Optimizasyonu

[*] Sivas Cumhuriyet University, Computer Engineering Department, 58140, Sivas, Turkey
[**] Sivas Cumhuriyet University, Software Engineering Department, 58140, Sivas, Turkey
Corresponding Author: Metin KALYON (kalyonmetin@hotmail.com)

1. INTRODUCTION

Optimization is the process of selecting the most effective solution to a difficulty depending on certain parameters (Emel and Taskin, 2003 Metaheuristic approaches, known for their effectiveness in addressing optimization challenges, are inspired by natural phenomena, social dynamics among species, and evolution principles (Öznur and Korukoğlu, 2003). Furthermore, these algorithms utilize exploration and exploitation phases to enhance optimization performance. The exploration phase conducts a global search over the entire solution space to discover diverse candidate regions, whereas the exploitation phase focuses on locally refining the search within the promising regions identified during exploration. (Albadr et al., 2020). Metaheuristic optimization techniques find applications in various fields, including finance (Emel and Taşkın, 2003), economics (Sharma and Kaur, 2021), energy (Rashedi et al., 2009), planning (Datta et al., 2011), image processing (Chouhan et al., 2018), and engineering design (Eyüp and Tanyıldızı, 2018). In this study, 5 metaheuristics developed in recent years (Artificial Rabbit Optimization-ARO, Black Widow Optimization-BWO, Prairie Dog Optimization-PDO, Mountain Gazelle Optimization-MGO and Crayfish Optimization Algorithm-COA) success in engineering design problems was compared. The proposed metaheuristic algorithms were applied to six real-world engineering design problems (Tension/Compression Spring, Pressure Vessel, Welded Beam, Speed Reducer, Gear Set, and Three-Bar Truss) each characterized by distinct design constraints, to determine their optimal design variables.

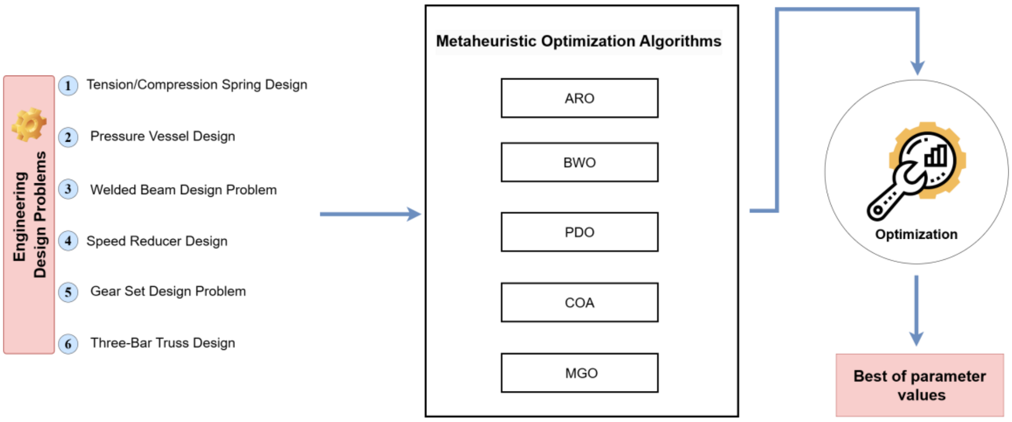


***Figure 1:** The general methodology of the study*

The general contributions of the study are as follows:

- To the best of our knowledge, this is the first study to compare BWO (2020), ARO (2022), PDO (2022), MGO (2022), COA (2023) algorithms for solving different engineering problems.
- To evaluate the performance of the metaheuristic algorithms, six well-known engineering design problems were selected: The Tension/Compression Spring, Pressure Vessel, Welded Beam, Speed Reducer, Gear Set, and Three-Bar Truss design problems.
- The experimental findings reveal that the algorithm demonstrating the highest level of success is ARO, BWO shows a very close performance to ARO and these two algorithms are followed by MGO, COA and PDO performances, respectively, while Among the algorithms used, PDO exhibited the lowest performance.

The remainder of this paper is organized as follows. Section 2 describes the five metaheuristics (ARO, BWO, PDO, MGO, COA) and formalizes the six engineering design problems together with their constraints. Section 3 details the experimental setup—common budget and stopping rule, hardware/software environment—and reports the simulation results with convergence profiles. Section 4 presents the discussion and literature comparison, benchmarking our three best-performing algorithms against recent baselines under identical parameters. Finally, Section 5 concludes the study and discusses possible avenues for future research.

2. METHODOLOGY

2.1. Metaheuristic optimization algorithms

2.1.1. Artificial Rabbit Optimization (ARO)

ARO is a nature-inspired metaheuristic proposed in 2022 that models rabbits' foraging, random hiding, and energy-budgeting behaviours (Wang et al., 2022). In ARO, the exploration phase corresponds to indirect foraging—rabbits search in the neighbourhoods of other rabbits rather than around their own location—whereas the exploitation phase mirrors random concealment, i.e., constructing/choosing nearby nests to refine promising regions (Wang et al., 2022; Bakır, 2024). The energy state governs the transition from exploration to exploitation across iterations. ARO has been applied in diverse domains, including water productivity prediction of solar still designs (Alsaiari et al., 2023), demand-response-based microgrid energy management (Alamir et al., 2023), load-frequency control in isolated microgrids (Khalil et al., 2023), and LSTM-based stock price forecasting (Gülmez, 2023). In implementation, candidate solutions represent rabbits and are updated via selection, indirect-foraging moves, random hiding (nest selection), and energy-driven phase switching.

In ARO, the transition from the exploration phase to the exploitation phase is governed by the gradual reduction of the rabbit's energy at each iteration level (Gülmez, 2023). The core mechanisms of the ARO algorithm are outlined in Table 1 below (Bakır, 2024).

Table 1: Pseudo-code of the ARO Algorithm

İnitiation
Create a rabbit population randomly, P
for i = 1 : N (rabbit population in the population) **do**
Measure each rabbit's suitability value, fi
end
repeat
Selection phase
Select P initial rabbits
Search phase
Meandering foraging (Exploration)
Random storage (Exploitation)
Update phase
Get the new population count P
until the termination conditions are met
end

2.1.2. Black Widow Optimization (BWO)

BWO is a society-centric algorithm derived from the reproductive habits of black widow spiders (Hayyolalam and Kazem, 2020). BWO is used to solve data exchange problems, particularly in mechanical engineering, electrical and electronics engineering, civil engineering, energy engineering, industrial engineering, and research fields such as image processing, networking, environment, robotics, planning and programming, and healthcare (Bektaş and

Serteller, 2023). In general terms, this algorithm models the mating behavior and post-mating cannibalism observed in black widow spiders. To put it differently, it embodies concepts from Darwin's evolutionary theory, particularly the survival of the strongest and the dominance of the most capable. Specifically, variations within a spider population point to genetic alterations (Hu et al., 2022). BWO comprises four phases: population initialization, reproduction, cannibalization, and mutation. The procedures of the BWO are outlined in Table 2 (Kuyu, 2023).

Table 2: Pseudo-code of the BWO Algorithm

1	Set the algorithm's initial parameters
2	Create the starting population of spiders
3	Determine the fitness value of the population
4	Have the stop criteria been met?
	a. Yes:
	- Determine the best spider as the solution
	b. Not acceptable:
	- Refresh the population
	- **Go to** step 2

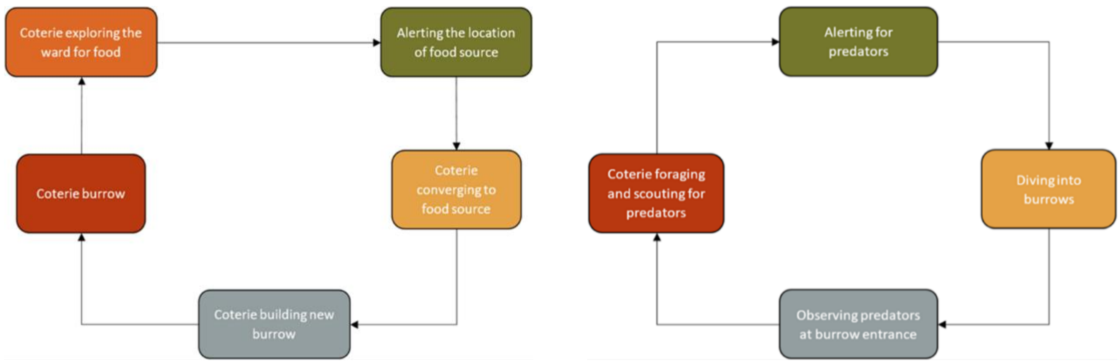
2.1.3. Prairie Dog Optimization (PDO)

PDO is an optimization algorithm inspired by the food-seeking and nest-building behaviors of prairie dogs (Ezugwu et al., 2022). Algorithm is used for optimization in applications such as efficient task scheduling in cloud computing (Hussein et al., 2023) and advanced control of hybrid microgrids, better transient response, and improved power quality (Sahoo et al., 2023). In PDO, the food-seeking and nest-building activities of prairie dogs are the exploration phase for optimization. One of the most important characteristics of prairie dogs is their communication skills. With these communication skills, they have sounds such as predator threats and various information about food. The exploitation phase in PDO covers the communication of prairie dogs and their fight against predators. In PDO, the exploration and exploitation phases are applied depending on the number $maks_{iter}$ (Ezugwu et al., 2022). The mathematical representations of the exploration and exploitation phases are given in Equations (1) and (2), respectively.

$$iter < \frac{Maks_{iter}}{4} \text{ and } \frac{Maks_{iter}}{4} \leq iter < \frac{Maks_{iter}}{2} \tag{1}$$

$$\frac{Maks_{iter}}{2} \leq \text{ iter } \leq 3\frac{Maks_{iter}}{4} \text{ and } 3\frac{Maks_{iter}}{4} \leq \text{ iter } \leq max \tag{2}$$

The exploration and exploitation phases of PDO are shown in Figure 2 (Izci et al., 2024). PDs build tunnels around abundant food sources. Group members search for fresh food sources within their nest and alert other group members to their discovery by making various sounds. When the current food source is depleted and a new food source of the highest quality is discovered, they build new tunnels around the food source. Prairie dogs use distinct vocalizations to encode both predator threats and food availability, which coordinates foraging and defense. The communication skills of prairie dogs play a vital role in enabling them to secure sufficient nutrition and protect themselves from predators. They also perceive differences in food source quality, predators, and hunting patterns, and respond accordingly. These specific behavioral abilities allow prairie dogs to find better, or near-optimal, solutions (Sahoo et al., 2023).



Exploration phase Exploitation phase

Figure 2: *PDO's exploration and exploitation phase models*

2.1.4. Mountain Gazelle Optimization (MGO)

MGO is a metaheuristic algorithm inspired by the social life of wild mountain gazelles and their hierarchy (Abdollahzadeh et al., 2022). MGO is used in areas such as parameter estimation of single and double diode photovoltaic cell models (Abbassi et al., 2023), development of IIR system identification (Ekinci and Izci, 2023), and brain stroke classification (Alomoush et al., 2024). This algorithm considers four key points in the life of mountain gazelles: bachelor herds, maternity herds, territorial bachelor males, and migratory behavior in search of food (Khodadadi et al., 2023). In the MGO algorithm, each gazelle X_i may belong to a bachelor herd, a natal herd, or a territorial herd (Abbassi et al., 2023). The mathematical formulation of the MGO algorithm is presented as follows.

- **Territorial Solitary Males:** Upon reaching adulthood and physical maturity, male mountain gazelles establish individual territories. These territories are quite far apart and in this way mountain gazelles live territorially. Male mountain gazelles fight for dominance and control of the territory over female gazelles.

- **Maternity Herds:** Maternity herds are vital in the life cycle of mountain gazelles, serving as protective groups where females give birth and nurture strong offspring. Male gazelles can also play a role in the delivery of gazelles and young males trying to possess females.

- **Bachelor Male Herds:** Upon reaching maturity, male gazelles typically establish and defend territories to gain access to and control over female groups. At this stage, young male gazelles begin to challenge adult males for dominance over females, and aggressive encounters may occur. (Alomoush et al., 2024).

- **Migration to Search for Food:** Mountain gazelles regularly make long journeys to graze and find food sources. They make these journeys with the help of their high-speed running and jumping skills (Alomoush et al., 2024).

Table 3: Pseudo-code of the MGO Algorithm

Input: Population size N and maximum number of iterations T
Output: Gazelle location and fitness potential
Generate a random population using $X_i (i = 1,2, \dots, N)$
Calculate the fitness values of gazelles
While stopping condition not satisfied
    for (calculate Xi for each gazelle)
        Calculate territorial bachelor males
        Calculate natal herds
        Calculate bachelor male herds
        Calculate transition to foraging
        Territorial bachelor males, Maternity herds, Bachelor Calculate the fitness values of male herds and transition stages to foraging and add them to the field
    end for
    Sort the population in ascending order
    Update $best_{gazelle}$
    Save $best_{gazelle}$ at the maximum population
end while
Return $best_{gazelle}$

2.1.5. Crayfish Optimization Algorithm (COA)

The COA belongs to the class of nature-inspired metaheuristic algorithm that simulates the foraging strategies, summer vacation behaviors, and competitive interactions of crayfish. The foraging and competitive behavior phases correspond to the exploitation phase of the COA, whereas the summer resort phase represents its exploration phase. (Jia et al., 2023). COA is used in the weight optimization of oil-immersed power and distribution type transformers at different power levels (Baş and Güner, 2025) and in the diagnosis and treatment of Parkinson's disease (Bacanin et al., 2024).

In the COA algorithm, temperature serves as a key factor regulating the balance between the exploration and exploitation stages. The change in temperature affects the behavior of crayfish and allows them to pass to different stages. When $temperature \geq 30℃$, crayfish pass to the summer resort stage. Crayfish show ideal foraging behavior between $temperature < 30℃$ and $temperature > 20℃$. The amount of feeding of crayfish is also related to temperature. The amount of feeding increases in the ideal temperature range (Shikoun et al., 2024). Temperature is calculated using Equation 3 and crayfish intake using Equation 4 as follows (Jia et al., 2023).

$$\text{temperature} = \text{rand} \times 15 + 20 \tag{3}$$

$$p = C_1 \times \left(\frac{1}{\sqrt{2 \times \pi} \times \sigma)} \times \exp\left(-\frac{(\text{ temperature} - \mu)^2}{2\sigma^2} \right) \right) \tag{4}$$

where temperature is the ambient temperature of the crayfish, $rand$ is a uniform random number in (0,1). μ represents the optimum temperature for crayfish, σ and $C1$ are the parameters used to control crayfish intake at different temperatures. When $temperature \geq 30℃$, this temperature value is too high for crayfish. In this case, crayfish usually take shelter in shady and cool caves and summer vacation spots are provided (Daulat et al., 2024). This situation is described by Equation 5 (Jia et al., 2023).

$$X_{\text{shade}} = (X_G + X_L)/2 \tag{5}$$

X_G represents the most suitable position, X_L represents the current position. $temperature \leq 30$ is the optimal temperature for crayfish feeding. At this temperature, crayfish move toward the food source and, upon locating it, assess the size of the food. If the food is too large, the crayfish tears the food into pieces and eats it (Xiao et al., 2024). The position of the food is shown by Equation 6. as follows. The size of the food is denoted by Q and is calculated by Equation 7 as follows (Jia et al., 2023).

$$X_{\text{food}} = X_G \tag{6}$$

$$Q = C_3 \times \text{ rand } \times (\text{ food }_i / \text{ best }_{\text{food}}) \tag{7}$$

Size of food Q, C_3 is the largest representative food factor and is a fixed value and its value is 3. The best i, the best food, are the suitability values of the i th crayfish and the food, respectively.

2.2. Engineering Design Problems

Real-world engineering design problems are characterized by highly complex objective functions and numerous constraints, which make them difficult to solve. These problems are frequently encountered in industrial applications and interdisciplinary research, where efficient and robust optimization methods are essential. (Altay, 2022). In this study, the six most commonly used real-world engineering design problems are discussed and detailed information about them is included.

2.2.1. Tension/Compression Spring Design

The tension/compression spring design problem is a problem that aims to create a spring design with minimum weight (Arora, 2004). The tension/compression spring problem has three decision variables (Özbay and Özbay, 2021).

- d x_1 : Wire diameter
- D x_2 : Average coil diameter
- N x_3: Number of active coils

The schematic structure of the tension/compression spring design problem is shown in Figure 3. as follows.

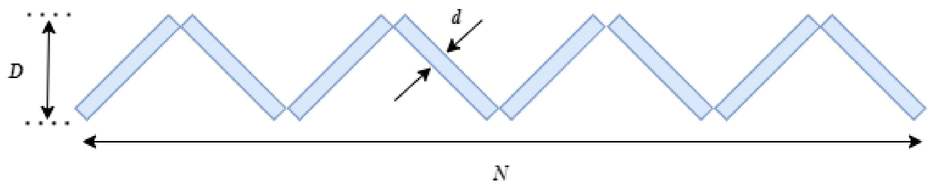


Figure 3: *Tension/Compression Spring Design*

Objective function(minimize):

$$f(x) = (x_3 + 2)x_2x_1^2 \quad (8)$$

Variable ranges:

$$0.05 \le x_1 \le 2 \quad (9)$$

$$0.25 \le x_2 \le 1.3 \quad (10)$$

$$2 \le x_3 \le 15 \quad (11)$$

2.2.2. Pressure Vessel Design

The objective of the pressure vessel design problem is to minimize the total cost of cylindrical pressure vessels (Jia et al., 2023). This design problem has four decision variables (Özbay and Özbay, 2021).

- T_s (X_1) : Thickness of the shell
- T_h (X_2) : Thickness of the head
- R (X_3) : Inner radius
- L (X_4) : Length of the cylindrical section without taking the head eye

The schematic structure of the pressure vessel design problem is shown in Figure 4. as follows.

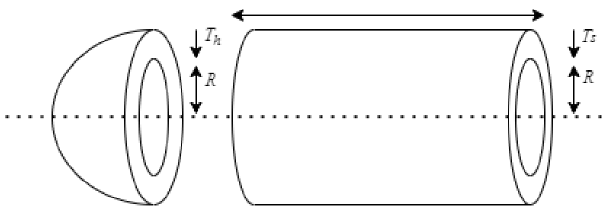


Figure 4: *Pressure Vessel Design*

The mathematical model of the problem is as follows (Jia et al., 2023). The model of the pressure vessel design problem with decision variables and constraints is shown below by Equation 12.

$$\vec{x} = [x_1 \quad x_2 \quad x_3 \quad x_4] = [T_S \quad T_h \quad R \quad L] \quad (12)$$

Objective function:

$$f(\vec{x}) = 0.6224x_1x_2x_3 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \quad (13)$$

Variable ranges:

$$\begin{gathered} 0 \le x_1 \le 99 \\ 0 \le x_2 \le 99.1 \\ 0 \le x_3 \le 200.1 \\ 0 \le x_4 \le 200 \end{gathered} \quad (14)$$

2.2.3. Welded Beam Design Problem

The main objective of the welded beam design problem is to produce a beam at minimum cost under certain constraints (He and Zhou, 2018). The problem consists of four design variables and five constraints. The design variables are as follows (Altay, 2022):

- h (x_1) : Weld thickness
- l (x_2) : Weld joint length
- t (x_3) : Beam width
- b (x_4) : Beam thickness

The mathematical model of the problem is as follows (Altay, 2022). The model of the welded beam design problem with decision variables and constraints is shown as follows by Equation 15.

$$\vec{x} = [x_1 \quad x_2 \quad x_3 \quad x_4] = [h \quad l \quad t \quad b] \tag{15}$$

Objective function:

$$f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) \tag{16}$$

Variable ranges:

$$0.125 \leq x_1 \leq 5 \text{ and } 0.1 \leq x_2, x_3, x_4 \leq 10 \tag{17}$$

2.2.4. Speed Reducer Design

The speed reducer design problem is a simple gearbox problem that allows the aircraft engine to rotate at the most efficient speed. This design problem aims to find the minimum cost weight of the speed reducer (Dhiman, 2021). The design variables are (Altay, 2022):

- $b\ x_1$: Face width
- $m\ x_2$: Tooth module
- $z\ x_3$: Number of teeth on the pinion
- $l_1\ x_4$: Length of the first shaft between bearings
- $l_2\ x_5$: Length of the second shaft between bearings
- $d_1\ x_6$: Diameter of the first shaft
- $d_2\ x_7$: Diameter of the second shaft

The mathematical model of the problem is as follows (Altay, 2022). The mathematical formulation of the speed reducer design problem, along with its decision variables and constraints, is expressed in Equation (18).

$$x = [x_1, x_2, x_3, x_4, x_5, x_6, x_7] = [b, m, z, l_1, l_2, d_l, d_2] \tag{18}$$

Objective function:

$$\begin{aligned} f(x) &= 0.785x_1x_2^2(3.333x_3^2 + 14.9334x_3 - 42.0934) \\ &-1.508x_1(x_6^2 + x_7^2) + 7.4777x_1(x_6^3 + x_7^3) + 1.508x_1(x_4x_6^2 + x_5x_7^2) \end{aligned} \tag{19}$$

Variable ranges:

$$\begin{aligned} &2.6 \leq x_1 \leq 3.6, 0.7 \leq x_2 \leq 0.8, 17 \leq x_3 \leq 28, 7.3 \leq x_4 \leq 8.3, \\ &7.3 \leq x_5 \leq 8.3, 2.9 \leq x_6 \leq 3.9, 5 \leq x_7 \leq 5.5 \end{aligned} \tag{20}$$

2.2.5. Gear Set Design Problem

The gear set design problem constitutes an unconstrained discrete optimization problem, aiming to identify the most effective gear parameters that satisfy performance objectives in the absence of explicit constraints. The goal of this problem is to minimize the gear ratio cost of the gear set. The design variables in this problem are the number of teeth of the gear wheels, T_a, T_b, T_d, T_f respectively. The mathematical model of the problem is as follows (Bakır, 2024). The design variable structure model of the gear set design problem is shown below by Equation 21.

$$\vec{x} = [x_1, x_2, x_3, x_4] = [T_a, T_b, T_d, T_f] \tag{21}$$

Objective function:

$$\text{Minimize } f_5(\vec{x}) = \left(\frac{1}{6.931} - \frac{T_b \cdot T_d}{T_a \cdot T_f}\right)^2 \tag{22}$$

Variable ranges:

$$x_1, x_2, x_3, x_4 \in \{12,13,14, \dots ,60\} \tag{23}$$

The schematic structure of the gear set design problem is shown in Figure 5. as follows (Kalyon and Arslan, 2024).

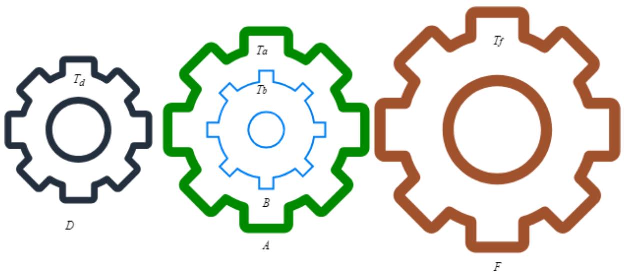


Figure 5: *Gear Set Design*

2.2.6. Three-Bar Truss Design Problem

The three-bar truss design problem is to minimize the weight of a statically loaded three-bar truss beam (Zhao et al., 2022). The design variables in this problem are the cross-sectional areas, x_1, x_2 (Özbay and Özbay, 2021). The mathematical model of the problem is as follows (Abdollahzadeh et al., 2021) The model of the design variable structure of the three-bar truss design problem is shown by Equation 24. as follows.

$$\vec{X} = [x_1 x_2] = [A_1 A_2] \tag{24}$$

Objective function:

$$f(\vec{X}) = \left(2\sqrt{2}X_1 + X_2\right) \times L \tag{25}$$

Variable ranges:

$$0 \leq x_i \leq 1, i = 1, 2. \tag{26}$$

Here; $l = 100cm, P = 2kN/cm^2, \sigma = kN/cm^2$

The schematic structure of the three-bar truss design problem is shown in Figure 6. as follows (Zhao et al., 2022).

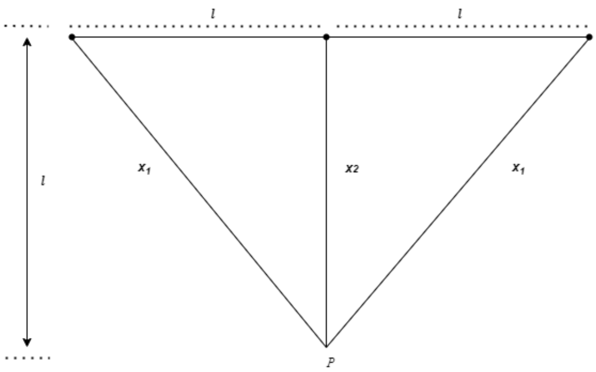


Figure 6: *Three-Bar Truss Design*

3. RESULTS

3.1. Experimental Design

The specific parameters and corresponding values of the algorithms employed in this study are presented in Table 4. The parameters P_P , C_R and P_M used in BWO are reproduction rate, cannibalization rate and mutation rate respectively. PDO parameters ρ is a food source alarm, ε is food source quality indicator and Δ is a Prairie dogs location difference. In COA, μ is the optimum temperature for feeding crayfish. σ and $\boldsymbol{C_1}$ are the parameters used to control the variation in food intake of crayfish at different temperatures. C_3 is the food factor representing the largest food. In addition to these control parameters, the metaheuristics also set the population size equal to 50 and the number of iterations equal to 1000 for a fair comparison. ARO and MGO do not have their own specific parameters, so they are not included in the table.

Table 4: Parameters

Control Parameters	**Algorithms**		
	BWO	PDO	COA
$\boldsymbol{P_P}$	0.60	-	-
$\boldsymbol{C_R}$	0.44	-	-
$\boldsymbol{P_M}$	0.40	-	-
$\boldsymbol{\rho}$	-	0.1	
$\boldsymbol{\varepsilon}$	-	$2.22E - 12$	
Δ	-	0.005	
$\boldsymbol{C_1}$	-	-	0.2
$\boldsymbol{C_3}$	-	-	3
$\boldsymbol{\mu}$	-	-	25
$\boldsymbol{\sigma}$	-	-	3

3.2. Simulation Results

Table 5 reports mean/std together with best and worst objective values, while Figs. 7–12 depict representative convergence trajectories. The following observations relate final solution quality (Table 5) and convergence behaviour (Figures 7–12), indicating both the attained value and the iteration range at which it is first reached.

In the table, the algorithms achieving the best mean and minimum (best) values are highlighted in bold. All methods were run under the same computational budget (population = 50, iterations = 1000; 50,000 function evaluations) and repeated for 100 independent runs (no early stopping). All experiments were performed using MATLAB R2024b on a workstation featuring an AMD Ryzen 7 8845HS processor and 32 GB of RAM.

Table 5: Simulation Results for Engineering Problems

Problem	Criteria	ARO	BWO	COA	MGO	PDO
Tension/ Compression Spring Design	mean	**1.27E-02**	1.32E-02	1.28E-02	1.28E-02	1.28E-02
	std	4.12E-07	8.70E-04	9.33E-05	1.54E-04	8.20E-05
	best	**1.27E-02**	**1.27E-02**	**1.27E-02**	**1.27E-02**	**1.27E-02**
	worst	1.27E-02	1.78E-02	1.32E-02	1.39E-02	1.32E-02
Pressure Vessel Design	mean	**5.89E+03**	6.51E+03	6.00E+03	6.45E+03	4.70E+04
	std	1.98E+03	4.36E+02	1.18E+02	5.16E+02	7.34E+04
	best	**5.89E+03**	5.89E+03	5.89E+03	5.89E+03	6.09E+03
	worst	5.89E+03	7.32E+03	6.48E+03	7.32E+03	3.05E+05
Welded Beam Design Problem	mean	**1.72E+00**	1.96E+00	1.73E+00	1.73E+00	2.19E+00
	std	3.79E-13	3.61E-01	2.79E-04	1.95E-02	6.33E-01
	best	**1.72E+00**	**1.72E+00**	**1.72E+00**	**1.72E+00**	1.74E+00
	worst	1.72E+00	3.77E+00	1.73E+00	1.89E+00	5.54E+00
Speed Reducer Design	mean	**2.99E+03**	3.04E+03	2.99E+03	**2.99E+03**	3.19E+03
	std	9.76E-10	2.50E+02	4.04E-01	1.32E-12	3.19E+01
	best	**2.99E+03**	**2.99E+03**	2.99E+03	**2.99E+03**	3.05E+03
	worst	2.99E+03	5.46E+03	3.00E+03	2.99E+03	3.24E+03
Gear Set Design Problem	mean	1.06E-16	**0.00E+00**	3.86E-15	5.51E-16	1.04E-20
	std	2.51E-16	0.00E+00	5.34E-15	1.77E-15	1.04E-19
	best	1.45E-21	0.00E+00	7.08E-21	1.11E-21	0.00E+00
	worst	1.71E-15	0.00E+00	2.94E-14	1.41E-14	1.04E-18
Three-Bar Truss Design Problem	mean	**2.64E+02**	**2.64E+02**	2.64E+02	2.64E+02	2.64E+02
	std	1.71E-13	1.72E-13	1.89E-04	4.92E-03	8.48E-02
	best	**2.64E+02**	**2.64E+02**	2.64E+02	**2.64E+02**	**2.64E+02**
	worst	2.64E+02	2.64E+02	2.64E+02	2.64E+02	2.65E+02

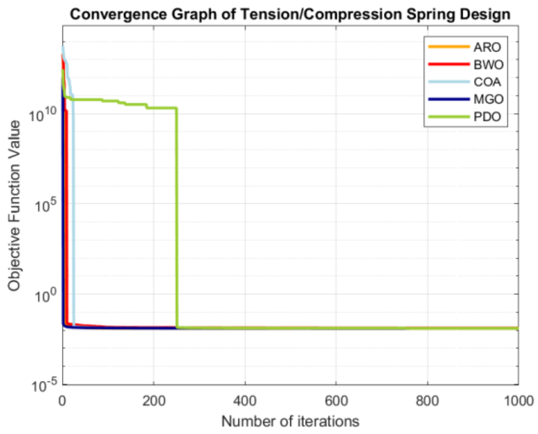


Figure 7: *Convergence Plot for Tension/Compression Spring Design Problem*

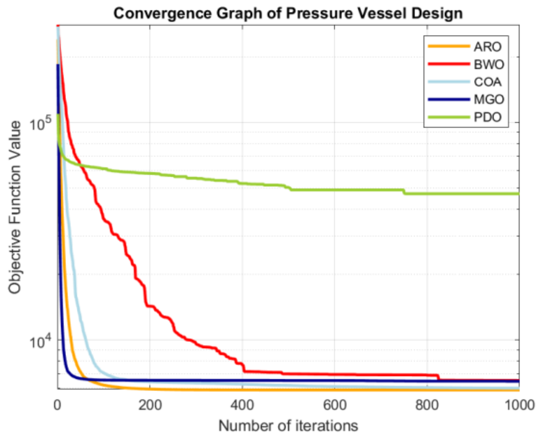


Figure 8: *Convergence Plot for Pressure Vessel Design*

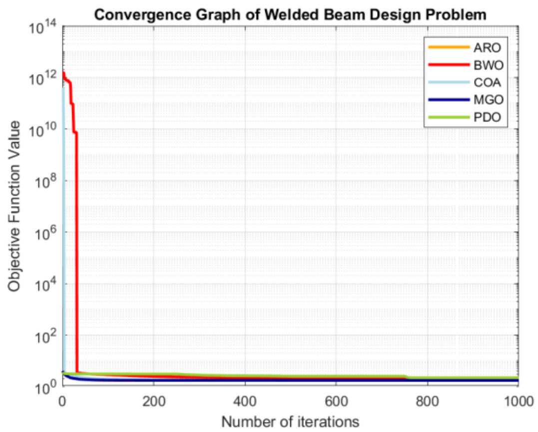


Figure 9: *Convergence Plot for Welded Beam Design Problem*

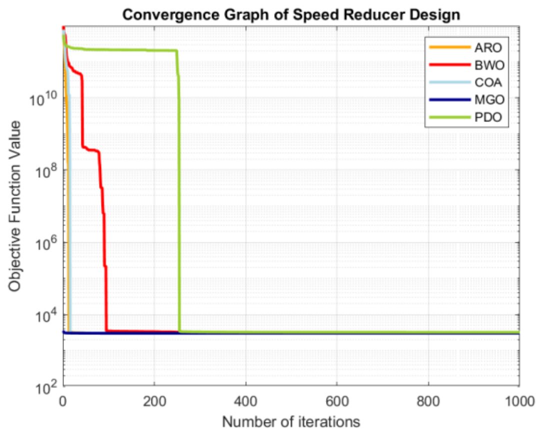


Figure 10: *Convergence Plot for Speed Reducer Design Problem*

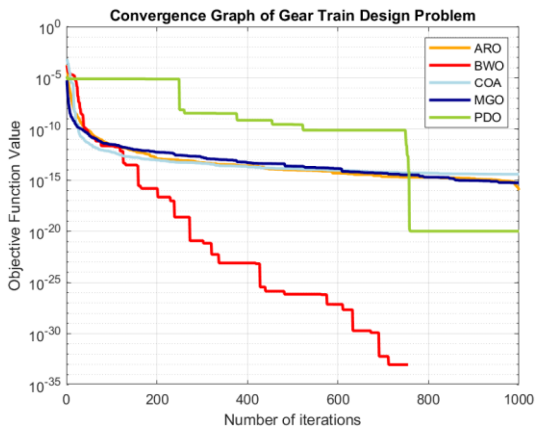


Figure 11: *Convergence Plot for Gear Set Design Problem*

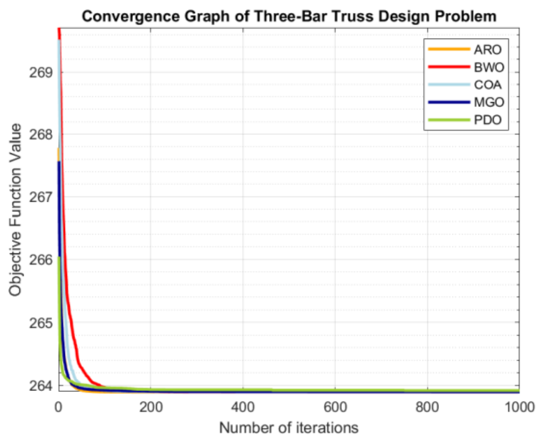


Figure 12: *Convergence Plot for Three-Bar Truss Design Problem*

- *Tension/Compression Spring:*

The optimal value f* = 1.27×10^-2 is attained by multiple algorithms. Consistent with Table 5, ARO yields the lowest mean (1.27×10^-2) with a negligible standard deviation (≈ 4.12×10^-7), indicating high stability. BWO, COA and MGO exhibit very similar means; PDO is slightly weaker.

Convergence profiles indicate that, except for PDO, the algorithms reach the vicinity of f* around iteration 30 and remain stable thereafter.

- *Pressure Vessel:*

The best value f* ≈ 5.89×10^3 is matched by ARO, BWO, COA and MGO. ARO achieves the lowest mean in Table 5, whereas PDO exhibits a much higher mean and large dispersion, reflecting instability on this constrained task.

Convergence profiles show MGO with the fastest early descent. ARO converges more gradually but stabilizes at a competitive final value; curves align within the same optimum neighbourhood in later iterations.

- *Welded Beam:*

The best value is approximately 1.72 and is achieved by ARO, BWO, COA and MGO. ARO provides the lowest mean with very small variance; COA and MGO are close. BWO presents a higher mean and wider spread.

Convergence profiles reveal rapid early descent for most methods; BWO settles later but ultimately attains the same terminal level.

- *Speed Reducer:*

The best value is approximately 2.99×10^3 and is obtained by ARO, COA, MGO and BWO. On mean performance, ARO, COA and MGO are nearly identical and lowest; BWO is higher and more variable, while PDO trails on both mean and best.

Convergence profiles cluster around the optimum band after roughly 250 iterations. BWO and especially PDO are slower in the early-to-mid phase; ARO, COA and MGO sustain the leading terminal level.

- *Gear Set:*

BWO consistently reaches the exact optimum f* = 0 with zero variance. PDO achieves f* in some runs and exhibits a near-zero mean (~10^-20). Other algorithms remain near zero (~10^-21 to 10^-16) but do not attain exact zero as consistently as BWO, mirroring the mean/worst statistics in Table 5.

Convergence profiles show a steep early drop for BWO, whereas other methods approach zero asymptotically.

- *Three-Bar Truss:*

Best and mean values concentrate around f* ≈ 2.64×10^2 for all algorithms. The principal difference is stability: ARO and BWO display near machine-precision variances (~10^-13), whereas COA, MGO and PDO exhibit larger dispersion; PDO shows the weakest worst case (~2.65×10^2).

Convergence profiles indicate that all methods reach similar function levels within the allotted budget; early differences diminish as iterations proceed.

Across the benchmark, ARO consistently attains lowest or equal-best objective values with small variability, BWO ranks second overall and dominates the Gear Set problem, while COA and MGO constitute a competitive middle tier. PDO is generally weaker and less stable, particularly on the Pressure Vessel problem.

4. DISCUSSION AND LITERATURE COMPARISON

To further strengthen the validity of our findings, we extended the comparative analysis by focusing on the three best-performing algorithms from our study (ARO, BWO, and COA) and benchmarking them against eight additional algorithms reported in Zhong et al. (2025), including the recently proposed SFOA together with well-established optimizers such as PSO, GWO, HHO, SSA, AOA, BBO, and WOA. This selection enables a comprehensive performance evaluation involving a total of eleven algorithms. Zhong et al. (2025) was deliberately chosen as the reference point because it applies the same identical computational budget (population = 50, iterations = 1000)as our study (population = 50, iterations = 1000), thereby ensuring a fair comparison, and because it represents one of the most recent and comprehensive contributions in the field. Since the gear set design problem was not included in Zhong et al. (2025), the comparison covers the remaining five problems.

Table 6. Comparative Results of Metaheuristics on the Tension/Compression Spring Design Problem (*: Values reproduced from Zhong et al. (2025); our results reported under the same budget (population = 50, iterations = 1000)**)**

Algorithm	x_1	x_2	x_3	f_{min}	rank
AOA*	0.05	0.31053	15	0.0132	9
ARO	0.05168	0.35651	11.3012	0.01267	2
BBO*	0.06218	0.66538	3.64333	0.01452	10
BWO	0.05169	0.35679	11.2845	0.01267	1
COA	0.05151	0.3524	11.5495	0.01267	3
GWO (Zhong et al., 2025).	0.05	0.31742	14.0323	0.01272	6
HHO*	0.05695	0.49701	6.15078	0.01314	8
PSO *	0.0521	0.36657	10.7338	0.01267	5
SFOA (Zhong et al., 2025).	0.0517	0.35686	11.2804	0.01267	4
SSA *	0.05	0.31384	14.514	0.01296	7
WOA (Zhong et al., 2025).	0.06422	0.73992	3.01467	0.0153	11

The results (Tables 6–10) clearly indicate that ARO, BWO, and COA not only maintain their leading performance within our study but also remain competitive with the advanced algorithms proposed in recent studies from the literature. In particular, ARO and BWO consistently match the best-known optima in the spring and welded beam design problems, while ARO and COA achieve superior or equivalent outcomes in the pressure vessel and speed reducer problems. In the three-bar truss problem, ARO and BWO perform on par with the top results of SFOA and other baselines. These findings confirm that the most recent metaheuristics studied here are not only effective in isolation but also competitive against the strongest algorithms currently available in the literature.

Table 7. Comparative Results of Metaheuristics on the Pressure Vessel Design Problem (*: Values reproduced from Zhong et al. (2025); our results reported under the same budget (population = 50, iterations = 1000)**)**

Algorithm	x_1	x_2	x_3	x_4	f_{min}	**rank**
AOA*	1.46173	0.85818	74.1756	200	26389.8	11
ARO	0.77817	0.38465	40.3196	200	5885.33	1
BBO*	0.89606	0.44322	46.3859	129.882	6124.83	7
BWO	0.77817	0.38465	40.3196	200	5885.33	3
COA	0.77963	0.38542	40.3851	199.091	5889.37	4
GWO (Zhong et al., 2025).	0.78903	0.39062	40.8735	192.451	5907.56	5
HHO*	1.09769	0.53892	56.491	53.9482	6696.61	9
PSO *	0.99905	0.49383	51.7643	84.936	6380.2	8
SFOA (Zhong et al., 2025).	0.77817	0.38465	40.3196	200	5885.33	2
SSA *	0.87077	0.43042	45.1175	142.603	6065.9	6
WOA (Zhong et al., 2025).	1.25891	0.62453	65.2252	10	7336.52	10

Table 8. Comparative Results of Metaheuristics on the Welded Beam Design Problem (*: Values reproduced from Zhong et al. (2025); our results reported under the same budget (population = 50, iterations = 1000)

Algorithm	x_1	x_2	x_3	x_4	f_{min}	**rank**
AOA*	0.18722	5.59408	10	0.21062	2.20208	9
ARO	0.20573	3.47049	9.03662	0.20573	1.72485	4
BBO*	0.38762	2.04817	6.55948	0.39046	2.31741	10
BWO	0.20573	3.47049	9.03662	0.20573	1.72485	5
COA	0.20571	3.47092	9.03663	0.20573	1.72489	6
GWO (Zhong et al., 2025).	0.20434	3.28074	9.03588	0.20579	1.69728	3
HHO*	0.125	5.79827	8.95883	0.21044	1.89584	8
PSO *	0.20573	3.25312	9.03662	0.20573	1.69526	2
SFOA (Zhong et al., 2025).	0.20572	3.25321	9.03661	0.20573	1.69526	1
SSA *	0.17274	4.03835	9.03244	0.20592	1.74724	7
WOA (Zhong et al., 2025).	0.34779	1.96759	8.04843	0.3644	2.51592	11

The higher standard deviation values obtained by PDO, particularly in the Pressure Vessel problem, may be associated with the stochastic nature of its communication and predator-fight mechanisms. Although these mechanisms contribute to enhancing the exploration capability of the algorithm, they may negatively affect convergence stability when solving nonlinear and highly constrained problems. As a result, the algorithm may exhibit a wider distribution around the optimum solution in some cases, which can lead to relatively lower performance compared to the other algorithms.

When the overall results are evaluated, ARO achieves better performance compared to the other algorithms. This can be explained by its adaptive search mechanism based on the detour foraging, random hiding, and energy shrink strategies. The detour foraging mechanism allows individuals to explore different regions of the search space, especially in the early iterations, which helps prevent premature convergence and improves global search capability. As the iterations progress, the search becomes more focused. Meanwhile, the random hiding mechanism enhances the local search process by allowing individuals to concentrate around promising regions, thereby improving solution accuracy and contributing to more stable convergence. In addition, the energy shrink mechanism provides a natural transition between exploration and exploitation, where exploration is dominant in the early stages and exploitation becomes more effective in the later stages. As a result, ARO maintains a good balance between exploration and exploitation, making it more effective and stable, particularly for solving nonlinear and constrained engineering design problems.

Table 9. Comparative Results of Metaheuristics on the Speed Reducer Design Problem
(*: Values reproduced from Zhong et al. (2025); our results reported under the same budget (population = 50, iterations = 1000)**)**

Algorithm	x_1	x_2	x_3	x_4	x_5	x_6	x_7	f_{min}	rank
AOA*	3.6	0.7	17	7.3	8.3	3.558	5.316	3121.77	11
ARO	3.5	0.7	17	7.3	7.71532	3.35021	5.28665	2994.47	1
BBO*	3.5	0.7	17	7.3	7.8	3.35	5.287	2996.36	5
BWO	3.5	0.7	17	7.3	7.71532	3.35021	5.28665	2994.47	2
COA	3.50002	0.7	17	7.3	7.71576	3.35022	5.28666	2994.5	3
GWO (Zhong et al., 2025).	3.502	0.7	17	7.367	7.821	3.353	5.287	2999.09	6
HHO*	3.592	0.7	17	7.319	7.8	3.353	5.287	3033.45	8
PSO *	3.6	0.7	17	7.3	8.3	3.35	5.287	3046.71	9
SFOA (Zhong et al., 2025).	3.5	0.7	17	7.3	7.8	3.35	5.287	2996.35	4
SSA *	3.578	0.7	17	7.904	8.27	3.477	5.287	3076.68	10
WOA (Zhong et al., 2025).	3.528	0.7	17	7.3	7.8	3.35	5.287	3007.24	7

Table 10. Comparative Results of Metaheuristics on the Three-Bar Truss Design Problem
(*: Values reproduced from Zhong et al. (2025); our results reported under the same budget (population = 50, iterations = 1000)**)**

Algorithm	x_1	x_2	f_{min}	rank
AOA*	0.75106	0.5373	266.162	11
ARO	0.78868	0.40825	263.896	3
BBO*	0.80644	0.36022	264.119	10
BWO	0.78868	0.40825	263.896	4
COA	0.78866	0.40828	263.896	5
GWO (Zhong et al., 2025).	0.78794	0.41033	263.897	6
HHO*	0.7898	0.40508	263.897	7
PSO *	0.78863	0.40837	263.896	2
SFOA (Zhong et al., 2025).	0.78868	0.40825	263.896	1
SSA *	0.78711	0.41268	263.898	8
WOA (Zhong et al., 2025).	0.79495	0.39078	263.924	9

5. CONCLUSIONS

Engineering design problems are preferred when evaluating the performance of newly developed metaheuristic algorithms. These problems may involve multiple objectives and complex variables, along with kinematic requirements, manufacturing requirements, and various nonlinear constraints on performance. As the size and complexity of the problem increases, the success rate of the algorithms tends to decrease. Numerous metaheuristic optimization algorithms and their modified alternatives have been introduced in the literature to obtain efficient solutions to engineering design problems. In this study, the success of ARO, BWO, COA, MGO and PDO metaheuristic algorithms is demonstrated with their performances on the following engineering design problems, respectively; tension/compression spring design, pressure vessel design, welded beam design problem, speed reducer design, gear set design and three-bar truss design. The simulation results and convergence plots of the metaheuristic algorithms were obtained, and their performances were systematically compared. The most successful among these algorithms is ARO, followed by BWO. In future studies, it is planned to solve various engineering problems in the industrial field with these and other current metaheuristics.

CONFLICT OF INTEREST

The authors declare no conflicts of interest

AUTHOR CONTRIBUTIONS

The authors contributed equally to the conception, design, and preparation of this work. Metin Kalyon was responsible for the conceptualization, data collection, and initial drafting of the manuscript. Sibel Arslan played a significant role in the data analysis, interpretation of findings, and critical revision of the manuscript. Both authors carefully reviewed and approved the final version prior to submission.

REFERENCES

1. Abbassi, R., Saidi, S., Urooj, S., Alhasnawi, B. N., Alawad, M. A., & Premkumar, M. (2023). An Accurate Metaheuristic Mountain Gazelle Optimizer for Parameter Estimation of Single-and Double-Diode Photovoltaic Cell Models. *Mathematics*, 11(22), 4565.
2. Abdollahzadeh, B., Gharehchopogh, F. S., & Mirjalili, S. (2021). African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems. *Computers & Industrial Engineering, 158*, 107408
3. Abdollahzadeh, B., Gharehchopogh, F. S., Khodadadi, N., & Mirjalili, S. (2022). Mountain gazelle optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems. *Advances in Engineering Software*, 174, 103282.
4. Alamir, N., Kamel, S., Hassan, M. H., & Abdelkader, S. M. (2023). An effective quantum artificial rabbits optimizer for energy management in microgrid considering demand response. *Soft Computing*, *27*(21), 15741-15768.
5. Alomoush, W., Houssein, E. H., Alrosan, A., Abd-Alrazaq, A., Alweshah, M., & Alshinwan, M. (2024). Joint opposite selection enhanced Mountain Gazelle Optimizer for brain stroke classification. *Evolutionary Intelligence*, 1-19.
6. Alsaiari, A. O., Moustafa, E. B., Alhumade, H., Abulkhair, H., & Elsheikh, A. (2023). A coupled artificial neural network with artificial rabbits optimizer for predicting water productivity of different designs of solar stills. *Advances in Engineering Software*, *175*, 103315.

7. Altay, E. V. (2022). Gerçek dünya mühendislik tasarım problemlerinin çözümünde kullanılan metasezgisel optimizasyon algoritmalarının performanslarının incelenmesi. *International Journal of Innovative Engineering Applications, 6*(1), 65-74.
8. Arora, J. S. (2004). Introduction to optimum design. Elsevier.
9. B. Gülmez, "Stock price prediction with optimized deep lstm network ¨with artificial rabbits optimization algorithm," *Expert Systems with Applications*, vol. 227, p. 120346, 2023.
10. Bacanin, N., Petrovic, A., Jovanovic, L., Zivkovic, M., Zivkovic, T., & Sarac, M. (2024, January). Parkinson's disease induced gain freezing detection using gated recurrent units optimized by modified crayfish optimization algorithm. In *2024 5th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI)* (pp. 1-8). IEEE.
11. Baş, E., & Güner, L. B. (2025). Weight Optimization of Oil Type Transformer with Crayfish Optimization Algorithm. *Sinop Üniversitesi Fen Bilimleri Dergisi*, *10*(1), 1-28.
12. Bektaş, Y. & Serteller, N. F. O (2023). Gerçek Parametre Optimizasyonu İçin Kara Dul Örümceği Optimizasyon Algoritması. *International Journal of Advanced Natural Sciences and Engineering Researches*, 7(11), 78-86.
13. D. Datta, A. R. Amaral, and J. R. Figueira, "Single row facility layout problem using a permutation-based genetic algorithm," *European Journal of Operational Research*, vol. 213, no. 2, pp. 388–394, 2011.
14. Daulat, H., Varma, T., & Chauhan, K. (2024, April). Augmenting the Crayfish Optimization with Gaussian Distribution Parameter for Improved 111 Optimization Efficiency. *In 2024 International Conference on Cognitive Robotics and Intelligent Systems (ICC-ROBINS)* (pp. 462-470). IEEE.
15. Dhiman, G. (2021). ESA: a hybrid bio-inspired metaheuristic optimization approach for engineering problems. *Engineering with Computers, 37*, 323- 353.
16. E. Eyup and E. Tanyıldızı, "Güncel metasezgisel optimizasyon algoritmalarının performans karşılaştırılması," in *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*. IEEE, 2018, pp. 1–16
17. E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "Gsa: a gravitational search algorithm," *Information sciences*, vol. 179, no. 13, pp. 2232– 2248, 2009.
18. Ekinci, S., & Izci, D. (2023). Enhancing IIR system identification: Harnessing the synergy of gazelle optimization and simulated annealing algorithms. *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, 5, 100225.
19. Ezugwu, A. E., Agushaka, J. O., Abualigah, L., Mirjalili, S., & Gandomi, A. H. (2022). Prairie dog optimization algorithm. *Neural Computing and Applications*, 34(22), 20017-20065.
20. G. G. Emel and Ç. Taşkın, "Genetik algoritmalar ve uygulama alanları," *Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, vol. 21, no. 1, pp. 129–152, 2002.
21. G. Hu, B. Du, X. Wang, and G. Wei, "An enhanced black widow optimization algorithm for feature selection," *Knowledge-Based Systems*, vol. 235, p. 107638, 2022.
22. H. Bakır, "Dynamic fitness-distance balance-based artificial rabbits optimization algorithm to solve optimal power flow problem," *Expert Systems with Applications*, vol. 240, p. 122460, 2024.
23. He, X., & Zhou, Y. (2018). Enhancing the performance of differential evolution with covariance matrix self-adaptation. *Applied Soft Computing, 64*, 227-243.
24. Hussein, A. H. A., Sunil, G., Kotha, M., Alzubaidi, L. H., & Arunasree, B. (2023, November). Prairie Dog Optimization Based Efficient Task Scheduling in the Cloud Computing. In *2023 International Conference on Integrated Intelligence and Communication Systems (ICIICS)* (pp. 1-5). IEEE.
25. Izci, D., Ekinci, S., & Hussien, A. G. (2024). Efficient parameter extraction of photovoltaic models with a novel enhanced prairie dog optimization algorithm. Scientific Reports, 14(1), 7945.

26. Jia, H., Rao, H., Wen, C., & Mirjalili, S. (2023). Crayfish optimization algorithm. *Artificial Intelligence Review*, 56(Suppl 2), 1919-1979.
27. Kalyon, M., & Arslan, S. (2024). Comparison of Black Widow Optimization and Aquila Optimizer with Current Metaheuristics. *International Journal of Multidisciplinary Studies and Innovative Technologies*, *8*(1), 17-25.
28. Khalil, A. E., Boghdady, T. A., Alham, M. H., & Ibrahim, D. K. (2023). Enhancing the conventional controllers for load frequency control of isolated microgrids using proposed multi-objective formulation via artificial rabbits optimization algorithm. *IEEE Access*, *11*, 3472-3493.
29. Khodadadi, N., El-Kenawy, E. S. M., De Caso, F., Alharbi, A. H., Khafaga, D. S., & Nanni, A. (2023). The Mountain Gazelle Optimizer for truss structures optimization. *Applied Computing and Intelligence*, 3(2), 116-144.
30. Kuyu, Y. Ç. (2023). *Optimizasyon Problemleri Için Yeni Metasezgisel Yaklaşımlar* (Doctoral dissertation, Bursa Uludag University (Turkey)).
31. L. Wang, Q. Cao, Z. Zhang, S. Mirjalili, and W. Zhao, "Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems," *Engineering Applications of Artificial Intelligence*, vol. 114, p. 105082, 2022.
32. M. A. Albadr, S. Tiun, M. Ayob, and F. Al-Dhief, "Genetic algorithm based on natural selection theory for optimization problems," *Symmetry*, vol. 12, no. 11, p. 1758, 2020
33. M. Sharma and P. Kaur, "A comprehensive analysis of nature-inspired meta-heuristic techniques for feature selection problem," *Archives of Computational Methods in Engineering*, vol. 28, pp. 1103–1127, 2021
34. Özbay, F. A., & Özbay, E. Martı Optimizasyon Algoritmasının Kısıtlı Mühendislik Tasarım Problemleri İçin Performans Analizi. *Adıyaman Üniversitesi Mühendislik Bilimleri Dergisi, 8*(15), 469-485
35. Öznur and S. Korukoğlu, "Genetik algoritma yaklaşımı ve yöneylem ¨ arastırmasında bir uygulama," *Yonetim ve Ekonomi Dergisi ¨*, vol. 10, no. 2, pp. 191–208, 2003.
36. S. S. Chouhan, A. Kaul, and U. P. Singh, "Soft computing approaches for image segmentation: a survey," *Multimedia Tools and Applications*, vol. 77, no. 21, pp. 28 483–28 537, 2018
37. Sahoo, G. K., Choudhury, S., Rathore, R. S., & Bajaj, M. (2023). A novel prairie dog-based meta-heuristic optimization algorithm for improved control, better transient response, and power quality enhancement of hybrid microgrids. *Sensors*, *23*(13), 5973.
38. Shikoun, N. H., Al-Eraqi, A. S., & Fathi, I. S. (2024). BinCOA: An Efficient Binary Crayfish Optimization Algorithm for Feature Selection. *IEEE Access, 12*, 28621-28635.
39. V. Hayyolalam and A. A. P. Kazem, "Black widow optimization algorithm: a novel meta-heuristic approach for solving engineering optimization problems," *Engineering Applications of Artificial Intelligence*, vol. 87, p. 103249, 2020.
40. Xiao, B., Wang, R., Deng, Y., Yang, Y., & Lu, D. (2024, March). Simplified Crayfish Optimization Algorithm. *In 2024 IEEE 7th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)* (Vol. 7, pp. 392-396). IEEE.
41. Y. Wang, Y. Xiao, Y. Guo, and J. Li, "Dynamic chaotic oppositionbased learning-driven hybrid aquila optimizer and artificial rabbits optimization algorithm: framework and applications," *Processes*, vol. 10, no. 12, p.2703, 2022.
42. Zhao, W., Wang, L., & Mirjalili, S. (2022). Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications. *Computer Methods in Applied Mechanics and Engineering, 388*, 114194.
43. Zhong, C., Li, G., Meng, Z., Li, H., Yildiz, A. R., & Mirjalili, S. (2025). Starfish optimization algorithm (SFOA): a bio-inspired metaheuristic algorithm for global optimization compared with 100 optimizers. *Neural Computing and Applications*, *37*(5), 3641-3683.