Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.050523

ARTICLE

African Bison Optimization Algorithm: A New Bio-Inspired Optimizer with Engineering Applications

Jian Zhao[1,2,*], Kang Wang[1,2], Jiacun Wang[3,*], Xiwang Guo[4] and Liang Qi[5]

[1]School of Science, University of Science and Technology Liaoning, Anshan, 114051, China

[2]Team of Artificial Intelligence Theory and Application, University of Science and Technology Liaoning, Anshan, 114051, China

[3]Department of Computer Science and Software Engineering, Monmouth University, West Long Branch, NJ 07764, USA

[4]Information and Control Engineering College, Liaoning Petrochemical University, Fushun, 113000, China

[5]Department of Computer Science and Technology, Shandong University of Science and Technology, Qingdao, 266590, China

*Corresponding Authors: Jian Zhao. Email: zhao@ustl.edu.cn; Jiacun Wang. Email: jwang@monmouth.edu

Received: 08 February 2024 Accepted: 21 August 2024 Published: 15 October 2024

ABSTRACT

This paper introduces the African Bison Optimization (ABO) algorithm, which is based on biological population. ABO is inspired by the survival behaviors of the African bison, including foraging, bathing, jousting, mating, and eliminating. The foraging behavior prompts the bison to seek a richer food source for survival. When bison find a food source, they stick around for a while by bathing behavior. The jousting behavior makes bison stand out in the population, then the winner gets the chance to produce offspring in the mating behavior. The eliminating behavior causes the old or injured bison to be weeded out from the herd, thus maintaining the excellent individuals. The above behaviors are translated into ABO by mathematical modeling. To assess the reliability and performance of ABO, it is evaluated on a diverse set of 23 benchmark functions and applied to solve five practical engineering problems with constraints. The findings from the simulation demonstrate that ABO exhibits superior and more competitive performance by effectively managing the trade-off between exploration and exploitation when compared with the other nine popular metaheuristics algorithms.

KEYWORDS

Optimization; metaheuristics; African bison optimization; engineering problems

1 Introduction

There are lots of complex computational problems in the fields of scientific computing, socioeconomics, and engineering which exhibit a high degree of non-linearity, non-differentiability, discontinuity, and high complexity [1]. In many cases, the conventional optimization algorithms that rely on analysis are inadequate for solving these problems. As a result, many new metaheuristic optimization algorithms are explored to overcome the above problems.

Nature has evolved over billions of years through genetic evolution to perfectly demonstrate its efficiency and magic. Meanwhile, humans have learned a great scale from natural systems to develop new algorithms and models for solving complex problems [2–4]. Nature-inspired metaheuristic optimization algorithms can be categorized into four groups, biological evolution-based, natural physics-based, human-based, and biological swarm-based approaches (See Fig. 1).

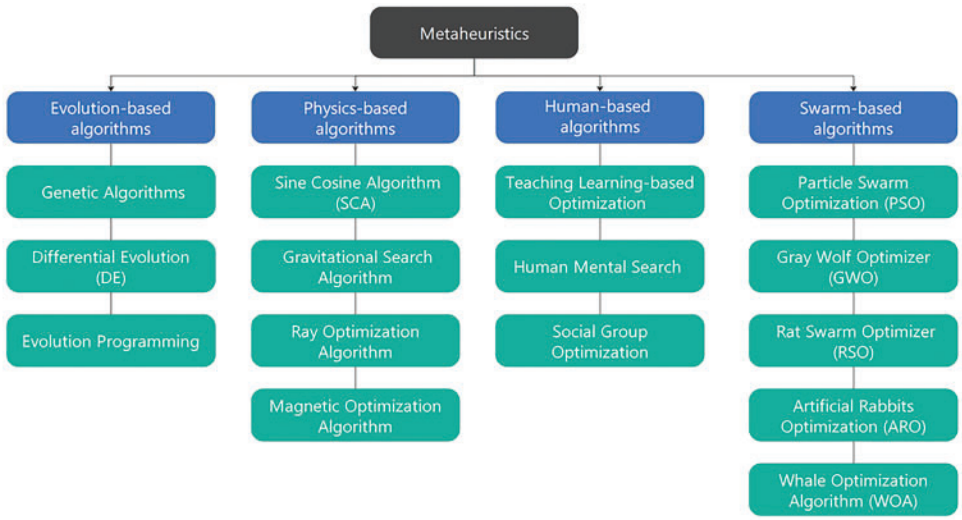

Figure 1: Metaheuristics classification

Evolution-based algorithms mimic natural evolutionary processes, including reproduction, mutation, recombination, and selection. Genetic Algorithms [5] stand out as a popular technique that draws inspiration from Darwin's theory of evolution and simulates its principles. Other popular algorithms are Differential Evolution (DE) [6] and Evolution Programming [7].

Physics-based algorithms are derived from fundamental principles of the natural world. These physical phenomena are represented by objects such as sound, light, force, electricity, magnetism, and heat. The most popular algorithm is the Sine Cosine Algorithm (SCA) [8]. Other popular algorithms are Gravitational Search Algorithm [9], and Magnetic Optimization Algorithm [10].

Human-based algorithms draw inspiration from a variety of human activities, including cognition and physical behaviors. There are teaching-learning based optimization [11], Human Mental Search [12], and Social Group Optimization [13].

Swarm-based algorithms emulate the social behaviors exhibited by animals. Among these techniques, Particle Swarm Optimization (PSO) [14], Whale Optimization Algorithm (WOA) [15], and Gray Wolf Optimizer (GWO) [16] have gained tremendous popularity. In recent years, new algorithms such as Flower Pollination Algorithm (FPA) [17], Chimp Optimization Algorithm (CHOA) [18], Rat Swarm Optimizer (RSO) [19], Tunicate Swarm Algorithm (TSA) [20], Reptile Search Algorithm (RSA) [21], and Artificial Rabbits Optimization (ARO) [22] are proposed.

All swarm-based optimization algorithms share common properties: exploration and exploitation. The tradeoff between exploration and exploitation has been a central issue in the field of evolutionary computation and optimization research. It is crucial for minimizing computational costs and achieving efficient results. Therefore, optimization algorithms must possess the capability to explore the global space and exploit local regions. The No Free Lunch (NFL) states that there is no algorithm that can solve all optimization problems well [23]. In other words, an algorithm can have brilliant results in a specific class of problems but fails to solve other problems.

The above-mentioned facts inspire us to propose a novel algorithm with swarm intelligence characteristics for solving optimization problems. In this paper, the proposed African Bison Optimization (ABO) algorithm is inspired by the survival behaviors of African bison, including foraging, jousting, mating, and eliminating from the herd. As far as we know, these behaviors are proposed for the first time, and there is no similar study in the literature. The African Buffalo Optimization algorithm models the organizational skills of African buffalos by two basic sounds, waaa and maaa when they are finding food sources [24]. Although the African buffalo and bison are related species, ABO proposed in this paper is based on completely different principles.

The main contributions of this paper can be summarized as follows:

- A swarm-based African Bison Optimization (ABO) is proposed and the survival strategies of bison are investigated and modeled mathematically.
- The proposed ABO is implemented and tested on 23 benchmark test functions.
- The performance of ABO is compared with some classical and latest optimizers.
- The efficiency of ABO is examined for solving the real-world engineering problems with constraints.

The rest of the paper is arranged as follows: Section 2 shows the biological characteristics of the African bison and introduces the proposed ABO algorithm. In Section 3, the performance and efficiency of ABO are tested by using benchmark test functions. The results of the algorithm on five constrained real-world engineering problems are discussed in Section 4. Section 5 provides a comprehensive conclusion of the study, along with future research directions.

2 African Bison Optimization Algorithm

2.1 African Bison

African bison is a genus of animals in the even-toed ungulates, bovidae, with five subspecies. As shown in Fig. 2, it has characteristics of a broad chest, strong limbs, a large head, and long horns. The African bison is widely distributed throughout most of sub-Saharan Africa. Their habitats are extensive, including open grasslands, savannas with drinking water, and lowland rainforests.

African bison are social animals, and only old or injured individuals will be broken away from the herd. The strongest bison becomes the leader and enjoys the right to eat the best grains. African bison cannot live without water, so they are rarely far away from water. They often hide in the shade, or dip in pools or mud to keep their bodies at a cooler temperature. To drink and feed, African bison often live near water. When food and water are plentiful, they will soak their entire body in water and reduce activities significantly to avoid high temperatures. Jousting and Mating take place in the rainy season when the temperature is comfortable and the food is abundant. Parts of the survival behaviors of the African bison described above are shown in Figs. 3–5.

Figure 2: African Bison in nature

Figure 3: Foraging behavior

Figure 4: Bathing behavior

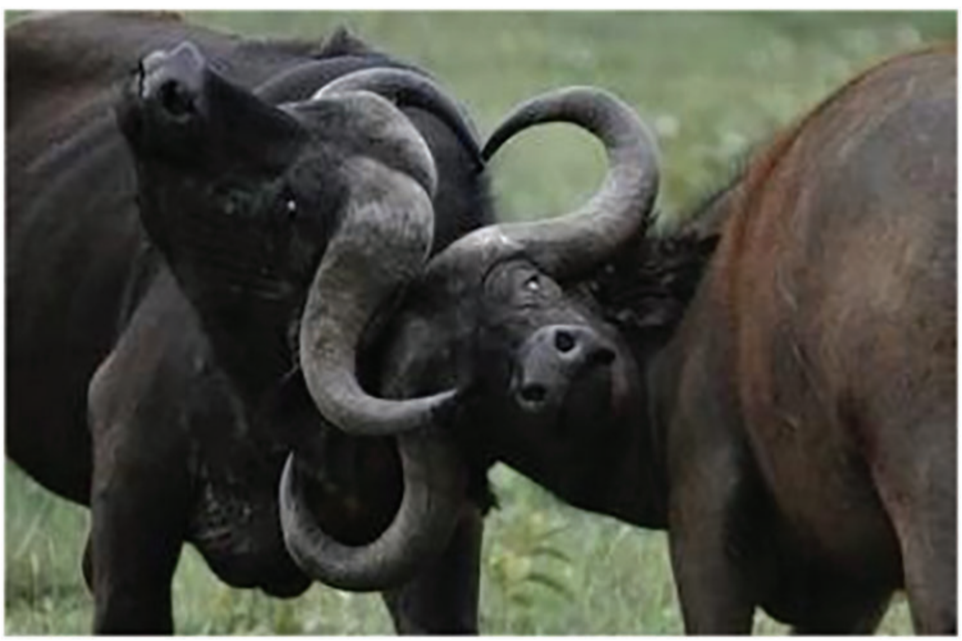

Figure 5: Jousting behavior

2.2 Mathematical Model and Algorithm

This subsection focuses on the mathematical modeling of foraging, jousting, mating, and eliminating of African bison herd, and provides a detailed description of ABO.

2.2.1 Initialization

In the initialization part, a uniformly distributed population is generated by random initialization, and this approach can be implemented by the following equation:

$$X_i = X_{min} + rand * (X_{max} - X_{min}) \tag{1}$$

where X_i is the position of i-th individual, $rand$ is a random number in the range from 0 to 1, and X_{min} and X_{max} are the lower and upper bounds of the problem, respectively.

In Eq. (2), the members of the bison population are represented through a population matrix. In this matrix, each row represents a potential solution, while the columns correspond to the suggested values for the variables of the problem.

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & \cdots & x_{1,d} \\ \vdots & \ddots & \vdots & & \vdots \\ x_{i,1} & \cdots & x_{i,j} & \cdots & x_{i,d} \\ \vdots & & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,j} & \cdots & x_{N,d} \end{bmatrix} \tag{2}$$

where X is the population matrix, N and d represent population size and individual dimension, respectively. $x_{i,j}$ is the j-th dimension of the i-th individual.

In ABO, each individual is metaphorically represented as a bison, serving as a potential solution to the given problem. The fitness values are determined by using a vector in Eq. (3).

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix} \tag{3}$$

where F_i is the fitness value of the i-th individual. F is the vector of these fitness values.

2.2.2 Cluster and Finding the Best Individual

In ABO, the population is divided into two subgroups with the same size $N/2$: male and female (See Fig. 6). In the herd, the position of the best individual also represents the place where the richest food source is located in. Therefore, we define the best position as the location of the richest food source and it is denoted as X_{Gbest} (See Fig. 7).

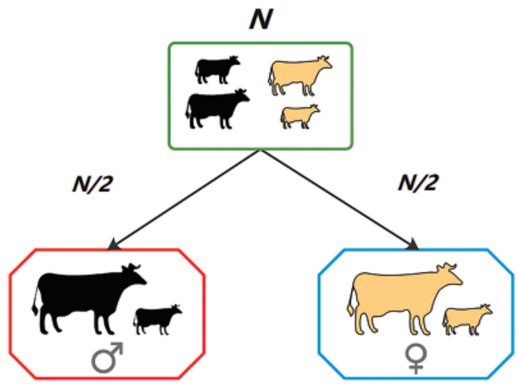

Figure 6: Examples of population grouping

Figure 7: The definition of the best individual

2.2.3 Defined the Satiety Rate and Temperature

African bison are very sensitive to temperature. When the temperature is high and food is plentiful, they soak their body in the water and reduce activities. They tend to breed during the rainy season when the temperature is suitable and food is plentiful. Individuals are competing for the right to mate. Inspired by above behaviors, we use the satiety rate (S) and temperature (Q) to balance the exploration and exploitation in ABO, and defined by Eqs. (4) and (5):

$$S = 2 * sin((3t/2T) * \pi) \tag{4}$$

$$Q = cos(-((T - t)/T) * \pi/2) \tag{5}$$

where the S is used for the balance of the exploration and exploitation phase, which has values from -2 to 2. Q is used for the balance of the development phase and it has values from 0 to 1. The current and maximum number of iterations are denoted by t and T, respectively.

2.2.4 Exploration Phase

If $|S| < \text{Threshold\_1}$ ($\text{Threshold\_1} = 0.5$), it means that the individuals are hungry. They will randomly update their position to find food. This behavior happens in the male and female subgroups and is modeled by Eqs. (6) and (7):

$$X_i^{t+1} = (X_{rand}^t * S) + A * ((X_{max} - X_{min}) * R_1) * R_2^3 \tag{6}$$

$$A = cos((F_{rand}/F_i) * \pi) * R_3 \tag{7}$$

where X^t_{rand} is the current position of a random individual, R_1 and R_2 are random numbers between [−1, 1], respectively. A represents the bison's ability to find food. F_{rand} and F_i are fitness values of random and current one, respectively. R_3 is a random number between [−5, 5].

2.2.5 Exploration Phase

If $|S| \geq$ Threshold_1, it means that food and water are enough. The bison will be bathing, jousting, mating, and eliminating according to the temperature level (Q).

If $Q \geq$ Threshold_2 (Threshold_2 = 0.6), it means that food and water are enough but the temperature is high. The bison will soak their body in water and reduce activities to avoid high temperatures. We define this behavior as bathing behavior. It is represented by Eq. (8) in the male and female subgroups:

$$X^{t+1}_i = 2 * (X^t_{Gbest} - X^t_i) * R_4 + exp(R_5 * Q^5) * cos(Q * 2\pi) \tag{8}$$

where X^t_{Gbest} is the best position in the herd, R_4 and R_5 are the random numbers between [0, 2], respectively.

If $Q <$ Threshold_2, it means that food and water are enough and the temperature is suitable for surviving. Then, the herd enters the jousting stage, and the winner has the right to mate.

The bison jousting behavior is represented by Eqs. (9) and (10) in the male and female subgroups:

$$J = X^t_{Sbest} - X^t_{Sbest} * \left(\frac{rand * X^t_i}{2}\right) * (cos(X^t_i) + sin(X^t_i)) * {R_6}^5 \tag{9}$$

$$X^{t+1}_i = X^t_{Sbest} - J * R_7 \tag{10}$$

where J is the fighting power of bison in the population. R_6 is a random number between [−0.01, 0.01], R_7 is a random number between [0, 2], and X^t_{Sbest} denotes the position of the best individual in the male or female subgroup.

The mating behavior of male bison is represented by Eqs. (11) and (12):

$$MM = cos(F_{i\_f}/F_{i\_m}) \tag{11}$$

$$X^{t+1}_{i\_m} = X^t_{i\_m} + sin(2\pi * R_8) * R_8 * MM * (X^t_{i\_m} - X^t_{i\_f}) \tag{12}$$

where MM is the mating ability of male individuals, F_{i_m} and F_{i_f} are the fitness values of the current male and female individual, respectively. $X^t_{i\_m}$ and $X^t_{i\_f}$ denote the current position in the male and female population, respectively. R_8 is a random number between [0, 1].

The mating behavior of female bison is achieved by the following expressions:

$$MF = sin(F_{i\_m}/F_{i\_f}) \tag{13}$$

$$X^{t+1}_{i\_f} = X^t_{i\_f} + cos(2\pi * R_9) * R_9 * MF * (X^t_{i\_f} - X^t_{i\_m}) \tag{14}$$

where MF is the mating ability of female individuals and R_9 is a random number between [0, 1].

The eliminating behavior means the aged or injured individuals will consciously detach themselves from the herd, thus allowing the entire population to remain at the highest level of competence. This behavior is represented by the following Eq. (15):

$$X^t_{worst} = X_{min} + R_{10} * (X_{max} - X_{min}) \tag{15}$$

where X^{t}_{worst} represents the eliminating of the old or injured individuals from the population. R_{10} is a random number between [0, 1].

2.2.6 The ABO Pseudo-Code

The above behaviors are mathematically modeled in this paper to construct ABO. To show the iterative process of ABO more clearly, Algorithm 1 gives the pseudo-code of the algorithm.

Algorithm 1: African Bison Optimization Algorithm

Input: X_{max}, X_{min}, $Popsize(N)$, $Max\_Iter(T)$, Iter(t)
Output: The best position and its fitness
Random initialization of population using Eq. (1)
The population is divided equally into two subgroups and calculate the fitness value for each individual
Find the best position and its fitness in two subgroups
For $t = 1:T$
Calculate the S and Q using Eqs. (4) and (5)
    If $(|S|) < Threshold\_1$
        Foraging behavior using Eq. (6) in two subgroups
    Else
        If $(Q < Threshold\_2)$
            Bathing behavior using Eq. (8) in two subgroups
        Else
            If $(r \geq 0.6)$
                Jousting behavior using Eq. (10) in two subgroups
            Else if $(0.1 < r < 0.6)$
                Mating behavior using Eqs. (12) and (14) in the male and female subgroup, respectively
            Else
                Eliminating behavior using Eq. (15) in two subgroups
            End if
        End if
    End if
Calculate the fitness value for each individual and update the best position and its fitness
End for
Return best solution

2.3 Computational Complexity

The complexity of an algorithm is determined by the population size (n), number of iterations (T), and problem dimensionality (d). The computational complexity of ABO can be summarized into the following three stages: the initialization of the population, the calculation of the fitness, and the position update of the individuals. It can be expressed as: O(ABO) = O(Initialization) + O(Function evaluation) + O(Location update of the individuals) = $O(n + Tn + Tnd)$.

2.4 Parameters Threshold_1 and Threshold_2 Sensitivity Analysis

The Threshold_1 and Threshold_2 are the key parameters to balance the exploration and exploitation stages of ABO. This section analyzes the sensitivity of the parameter Threshold_1 and Threshold_2 by setting different values. The results are shown in Table 1, where the Threshold_1 and Threshold_2 are set to (0, 0.4), (0.5, 0.6), (1, 0.8) and (2, 1), respectively. The performance of (0.5, 0.6) is the best compared with others.

Table 1: Results of different combinations of Threshold_1 and Threshold_2

Function		(0.5, 0.6)	(0, 0.4)	(1, 0.8)	(2, 1)
F1	Mean	**0**	0	0	0
F2		**0**	0	0	0
F3		**0**	0	0	0
F4		**0**	0	0	7.83439E-84
F5		1.090042539	7.359920014	**0.488914301**	0.70970151
F6		**0.029361478**	0.046890454	0.046474359	0.060522219
F7		0.000176978	**8.382E-05**	0.000136326	0.000210415
F8		−**12398.06721**	−12050.56684	−12320.76541	−11939.14066
F9		**0**	0	0	0
F10		**8.88178E-16**	8.88178E-16	8.88178E-16	8.88178E-16
F11		**0**	0	0	0
F12		**0.004475098**	0.032834546	0.020679249	0.014471592
F13		**0.046484731**	0.122087636	0.108531792	0.078069042
F14		1.233075861	2.676849143	1.330986282	**1.033865839**
F15		0.000344918	0.000685222	0.000338614	**0.000336355**
F16		−**1.031628435**	−1.031368619	−1.031627418	−1.031626074
F17		**0.397887361**	0.398676955	0.397887864	0.397887515
F18		3.000008251	3.002797211	3.000000625	**3.000000074**
F19		−**3.862781389**	−3.861817866	−3.862751303	−3.862769483
F20		−**3.319823751**	−3.235794903	−3.295627739	−3.31810832
F21		−**10.14934325**	−10.12086745	−10.14776529	−10.12883157
F22		−**10.40009336**	−10.36792913	−10.39813134	−10.38037045
F23		−**10.5320396**	−10.48339948	−10.52951159	−10.50306925

2.5 The Survival Behaviors Analysis

ABO is inspired by the survival behaviors of the African bison. The foraging behavior drives the individuals to search for local optimal solutions randomly. The bathing behavior enables the individuals to fully explore near the local optimal solutions, thus improving the quality of the solution. Through jousting and mating behavior, the population can produce better individuals, so that ABO can converge to the optimal value faster. The worst individual in the population is eliminated and replaced by the elimination behavior, so the whole population can always maintain diversity in the iteration.

3 Experimental Results and Discussion

This section presents the results of experiments conducted on 23 benchmark functions to demonstrate the performance superiority of ABO. The experimental results of ABO are compared with those of nine other metaheuristic optimization algorithms.

3.1 Benchmark Test Functions

To comprehensively evaluate the performance of ABO in terms of optimization, we test it on 23 benchmark functions. These functions can be classified into three main categories: Unimodal, Multimodal, and Fixed-dimension multimodal [25]. The presence of a single global optimum in Unimodal (F1–F7) benchmark functions serves as a valuable test for evaluating the exploration capability of the algorithm under consideration. The optimization algorithms' diversification capabilities are shown by using the Multimodal (F8–F13) and Fixed-dimension multimodal (F14–F23) benchmark functions. Details about these functions can be found in [15].

3.2 Experimental Setup

The parameters of the other algorithms (PSO, GWO, CHOA, SCA, RSA, FPA, WOA, RSO, and TSA) are derived from the corresponding papers. The experimentation process and algorithms have been carried out on an i5 processor operating at 2.60 GHz and 16 GB of RAM.

3.3 Performance Comparison

To ensure an equitable comparison, we maintain a consistent population size of 30 individuals across all optimization algorithms and perform 500 generations. The mean, std, best result, and ranking after 30 independent runs of each algorithm are presented in Tables 2–4, where the ranking is determined based on the mean.

Table 2: The mean, std, and best values on the unimodal functions

Functions		ABO	PSO	WOA	RSO	CHOA	FPA	GWO	RSA	SCA	TSA
F1	Mean	**0**	2.739504	4.96E-74	2E-259	1.62E-08	1863.119	6.53E-28	0	8.847723	3.1E-195
	Std	0	1.381861	1.66E-73	0	6.87E-08	645.9023	9.15E-28	0	13.03789	0
	Best	0	1.059479	1.53E-91	0	9.48E-28	1135.756	7.44E-29	0	0.097024	6.6E-205
	Rank	1	8	5	3	7	10	6	2	9	3
F2	Mean	**0**	4.677332	2.97E-51	1.4E-139	8.22E-07	30.14067	1.24E-16	0	0.026963	1.5E-100
	Std	0	1.170234	8.8E-51	5.1E-139	1.95E-06	5.886052	1.39E-16	0	0.042914	5.5E-100
	Best	0	3.016678	8.08E-60	0	6.14E-10	22.85537	1.56E-17	0	7.77E-05	1.3E-106
	Rank	1	9	4	3	7	10	6	2	8	3
F3	Mean	**0**	187.8865	45969.67	5.1E-255	102.1504	1456.598	1.28E-05	0	8696.327	1.4E-182
	Std	0	51.43875	12211.16	0	214.6503	348.5603	2.6E-05	0	5048.532	0
	Best	0	98.23863	16083.88	0	0.335248	885.9155	9.73E-09	0	1579.356	1.8E-187
	Rank	1	7	10	3	6	8	5	2	9	3
F4	Mean	**0**	1.924234	44.24386	4.69E-90	0.013704	23.59319	5.63E-07	0	35.77706	4.15E-92
	Std	0	0.229176	29.14158	2.57E-89	0.013756	3.252222	5.81E-07	0	14.17494	1.56E-91
	Best	0	1.49975	1.268282	0	0.000243	17.58065	2.58E-08	0	11.50938	2E-100
	Rank	1	7	10	4	6	8	5	2	9	3
F5	Mean	**1.090043**	806.674	27.79136	28.76958	28.93083	301659	26.98143	22.19044	24182.83	28.52099
	Std	4.29301	342.0505	0.443458	0.210932	0.065806	155153.4	0.796159	12.45145	61679.71	0.365193
	Best	0.000384	359.691	26.91447	28.14204	28.73378	78722.21	25.8111	2.76E-27	30.88447	28.05151
	Rank	1	8	5	6	7	10	4	2	9	6
F6	Mean	**0.029361**	2.29193	0.448323	3.278986	3.138063	1884.918	0.720921	6.946221	15.91793	5.941678

(Continued)

Table 2 (continued)

Functions		ABO	PSO	WOA	RSO	CHOA	FPA	GWO	RSA	SCA	TSA
	Std	0.036941	0.828593	0.257496	0.602236	0.388782	293.0147	0.37812	0.676677	22.48097	0.845228
	Best	1.19E-06	0.712171	0.097252	1.762142	2.336074	1262.862	7.1E-05	3.932309	4.294536	3.941324
	Rank	r1	5	2	6	6	10	3	8	9	7
F7	Mean	0.000177	18.89928	0.003832	0.00051	0.001878	0.324021	0.002313	0.000137	0.176878	**6.71E-05**
	Std	0.000166	15.2891	0.004263	0.000505	0.001714	0.087986	0.001023	0.000126	0.261323	6.2E-05
	Best	6E-06	1.562643	1.94E-05	3.59E-05	4.35E-05	0.180599	0.000859	4.17E-06	0.003958	2.19E-06
	Rank	3	10	7	4	5	9	6	2	8	1

Table 3: The mean, std, and best values on the multimodal functions

Functions		ABO	PSO	WOA	RSO	CHOA	FPA	GWO	RSA	SCA	TSA
F8	Mean	**−12398.1**	−6368.57	−10310.7	−5819.19	−5712.53	−6393.11	−5871.21	−5498.44	−3708.51	−3327.75
	Std	269.4774	1300.051	1587.465	897.3912	83.868	233.5901	719.9325	132.336	208.6469	461.9023
	Best	−12569.1	−8956.26	−12568.8	−6924.98	−6026.69	−6957.85	−7152.02	−5722.93	−4054.57	−4447.49
	Rank	1	5	3	6	7	4	6	8	9	10
F9	Mean	**0**	171.5526	5.68E-15	0	1.082867	183.3753	1.683353	0	35.99174	20.11077
	Std	0	32.82081	1.73E-14	0	2.984753	12.67538	2.497436	0	27.08043	42.84416
	Best	0	100.3091	0	0	3.98E-13	159.8033	5.68E-14	0	0.473392	0
	Rank	1	9	3	2	4	10	5	2	8	7
F10	Mean	**8.88E-16**	2.583473	4.91E-15	1.36E-15	19.96288	7.321006	1.09E-13	8.88E-16	10.44769	4.56E-15
	Std	0	0.347243	2.91E-15	1.23E-15	0.001377	1.265033	2.72E-14	0	9.624672	6.49E-16
	Best	8.88E-16	1.876971	8.88E-16	8.88E-16	19.9601	5.059116	7.55E-14	8.88E-16	0.075379	4.44E-15
	Rank	1	7	4	3	10	8	5	2	9	3
F11	Mean	**0**	0.130369	0	0	0.015213	18.85045	0.005716	0	0.812483	0.002264
	Std	0	0.060892	0	0	0.024037	4.806672	0.009628	0	0.379886	0.006266
	Best	0	0.053012	0	0	3.15E-13	11.20103	0	0	0.019209	0
	Rank	1	8	2	3	6	10	5	3	9	4
F12	Mean	**0.004475**	0.055503	0.022159	0.350741	0.315031	41.34128	0.043116	1.355855	19106.87	1.090577
	Std	0.005746	0.046017	0.010872	0.13331	0.106485	69.48961	0.01846	0.286829	71302.46	0.319172
	Best	1.37E-07	0.014218	0.007322	0.183195	0.148244	12.0622	0.013064	0.742027	0.920505	0.511953
	Rank	1	4	2	6	6	9	3	8	10	7
F13	Mean	**0.046485**	0.529532	0.558913	2.866925	2.809285	68964.71	0.717013	0.093335	307141.1	2.561741
	Std	0.051307	0.213334	0.248765	0.045647	0.10323	95611.47	0.30797	0.47412	882041.5	0.296405
	Best	2.66E-05	0.200496	0.099758	2.802804	2.586901	662.1132	0.101023	1.45E-30	2.805047	1.841649
	Rank	1	4	3	8	8	9	5	2	10	7

Table 4: The mean, std, and best values on the fixed-dimension multimodal functions

Functions		ABO	PSO	WOA	RSO	CHOA	FPA	GWO	RSA	SCA	TSA
F14	Mean	1.233076	2.770934	2.960349	2.381829	1.062489	**0.998144**	4.128514	3.671247	2.122922	10.36112
	Std	0.514499	2.623468	3.32548	1.847044	0.239231	0.000495	4.305079	2.755154	0.999311	5.126076
	Best	0.998004	0.998004	0.998004	0.998004	0.998004	0.998004	0.998004	1.01025	0.998004	1.992031
	Rank	3	6	7	5	2	1	9	8	5	10
F15	Mean	**0.000345**	0.000846	0.000683	0.001603	0.004891	0.000617	0.005057	0.001475	0.000941	0.00462
	Std	4.5E-05	0.000133	0.000423	0.003677	0.016176	0.000124	0.008592	0.000693	0.000353	0.008114
	Best	0.00031	0.000525	0.000309	0.000399	0.001234	0.000358	0.000308	0.00066	0.000322	0.000313
	Rank	1	4	3	7	9	2	10	7	5	8
F16	Mean	**−1.03163**	−1.03163	−1.03163	−1.03149	−1.02849	−1.03163	−1.03163	−1.02976	−1.03157	−1.02423

(Continued)

Table 4 (continued)

Functions		ABO	PSO	WOA	RSO	CHOA	FPA	GWO	RSA	SCA	TSA
	Std	5.6E-08	4.7E-16	2.11E-09	0.00018	0.009536	9.51E-08	2.35E-08	0.002971	4.2E-05	0.013599
	Best	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03161	−1.03163	−1.03163
	Rank	4	1	3	7	9	6	5	8	7	10
F17	Mean	**0.397887**	0.397887	0.397906	1.180402	0.400341	0.397891	0.397906	1.121332	0.400458	0.399368
	Std	1.05E-08	0	4.59E-05	1.215561	0.00516	4.89E-06	8.89E-05	1.276393	0.003794	0.003143
	Best	0.397887	0.397887	0.397887	0.398106	0.39789	0.397887	0.397887	0.399553	0.397951	0.397944
	Rank	2	1	6	10	8	4	5	10	9	7
F18	Mean	3.000008	**3**	3.00009	3.000051	3.000185	3	3.000044	5.811115	3.000176	12.26838
	Std	4.49E-05	6.32E-15	0.000185	0.000117	0.000193	4.12E-09	4.55E-05	8.577847	0.000516	22.61218
	Best	3	3	3	3	3.000007	3	3.000001	3	3	3.000001
	Rank	3	1	5	9	7	2	4	8	6	10
F19	Mean	**−3.86278**	−3.86278	−3.85399	−3.49464	−3.8547	−3.86278	−3.86162	−3.79781	−3.85453	−3.85952
	Std	1.19E-06	1.49E-14	0.013546	0.298841	0.002898	1.26E-08	0.002318	0.061135	0.002967	0.002388
	Best	−3.86278	−3.86278	−3.86278	−3.84748	−3.86221	−3.86278	−3.86278	−3.85726	−3.86177	−3.86265
	Rank	3	1	9	10	7	3	5	10	8	6
F20	Mean	**−3.31982**	−3.25859	−3.22231	−1.75724	−2.60286	−3.29848	−3.23749	−2.71951	−2.85248	−3.137
	Std	0.002542	0.060328	0.115741	0.492095	0.492071	0.015875	0.080339	0.382472	0.347177	0.124958
	Best	−3.32177	−3.322	−3.32106	−2.55937	−3.23866	−3.31623	−3.32199	−3.17246	−3.17695	−3.29692
	Rank	1	4	6	10	9	3	5	10	8	7
F21	Mean	**−10.1493**	−7.29673	−8.52957	−0.88619	−3.17522	−10.1399	−9.06434	−5.0552	−2.66714	−7.13858
	Std	0.00572	3.005022	2.529567	0.695632	2.041282	0.015535	2.520117	3.32E-07	1.688837	1.51868
	Best	−10.1532	−10.1532	−10.1523	−3.37099	−5.00764	−10.1531	−10.153	−5.0552	−5.11201	−9.70639
	Rank	1	6	5	10	9	2	4	8	10	7
F22	Mean	**−10.4001**	−9.60625	−7.33244	−0.93678	−3.06208	−10.2934	−10.2238	−5.08767	−2.95511	−6.26446
	Std	0.004205	1.956164	3.184573	0.513169	2.048768	0.151153	0.970056	7.99E-07	1.72394	2.116332
	Best	−10.4029	−10.4029	−10.4014	−2.07786	−5.03666	−10.4025	−10.4027	−5.08767	−6.8096	−9.97328
	Rank	1	5	6	10	9	3	4	8	10	7
F23	Mean	**−10.532**	−9.95552	−7.56613	−1.2381	−4.17562	−10.3441	−10.3559	−5.12847	−3.79545	−5.06172
	Std	0.010876	1.783603	3.098342	0.666509	1.652114	0.285731	0.978592	1.79E-06	1.473324	2.732771
	Best	−10.5364	−10.5364	−10.5362	−3.62608	−5.82866	−10.5233	−10.5363	−5.12848	−6.50974	−9.163
	Rank	1	5	6	10	9	4	3	7	10	8

According to Table 2, the mean, std, and best of ABO on F1–F4 are all zero, which indicates that the algorithm can accurately find the global optimum of these functions each time. For F5 and F6, ABO can achieve significantly better optimal solutions than other algorithms. On F7, ABO is similarly competitive, with optimization results next only to TSA and RSA. The results demonstrate that ABO has significant advantages in exploitation compared to other algorithms.

According to the experimental results in Tables 3 and 4, ABO achieves very competitive results. On F8–F13, F15, and F20–F23, the optimization results of ABO are better than other algorithms, and the ranking is 1. On the other functions, the proposed algorithm also achieves satisfactory results, and the optimization performance is better than most algorithms. These results illustrate that ABO has excellent performance in terms of exploration capability.

3.4 Convergence Analysis

Fig. 8 shows the convergence curves of ABO with the other algorithms on benchmark test functions. It can be seen that ABO has three different convergence behaviors. In the early stages, it explores fully the given search space, allowing the highest convergence efficiency. In the second stage, the algorithm initiates iterative processes aimed at approaching the optimal solution until reaching the maximum allowed number of iterations. The last stage is the iterative process of rapid convergence of ABO, where it finds the optimal solution to the benchmark test problem through full exploration and exploitation. The above analysis show that ABO achieves a good balance of exploration and exploitation than other metaheuristic optimization algorithms.

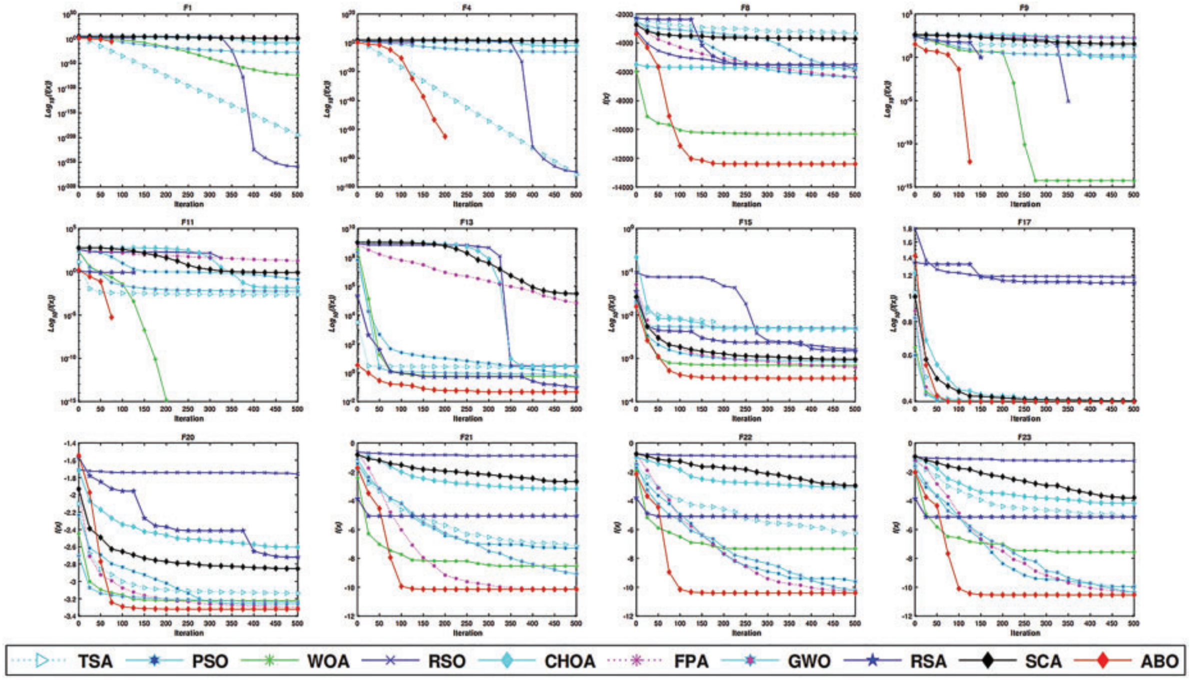

Figure 8: Convergence plots for various benchmark functions

3.5 Statistical Testing

In addition to utilizing standard statistical measures such as mean and standard deviation, we have also employed the Wilcoxon sum rank test for further analysis and comparison [26]. It is used to assess whether ABO's results statistically significantly differ from that of other competing algorithms. In this paper, the sample size of the Wilcoxon sum rank test is 30 and the confidence interval is 95%. The significance of an algorithm can be determined by examining its p-value. If the p-value of a algorithm is lower than 0.05, it signifies that the algorithm exhibits statistical significance when compared to the other algorithms under consideration. Table 5 shows the Wilcoxon sum rank test for the 23 benchmark test functions. From the experimental results of the table, it can be obtained that the p-value obtained from the comparison of ABO with the other algorithms is much less than 0.05 in general for all the benchmark functions. It can be shown that ABO has a significant difference compared with other algorithms.

Table 5: p-values of the Wilcoxon rank-sum test with 5% significance for F1–F23

Functions	PSO	WOA	RSO	CHOA	FPA	SCA	GWO	RSA	TSA
F1	1.21E-12	1.21E-12	1.46E-04	1.21E-12	1.21E-12	1.21E-12	1.21E-12	NaN	1.21E-12
F2	1.21E-12	1.21E-12	1.27E-05	1.21E-12	1.21E-12	1.21E-12	1.21E-12	NaN	1.21E-12
F3	1.21E-12	1.21E-12	5.58E-03	1.21E-12	1.21E-12	1.21E-12	1.21E-12	NaN	1.21E-12

(Continued)

Table 5 (continued)

Functions	PSO	WOA	RSO	CHOA	FPA	SCA	GWO	RSA	TSA
F4	1.21E-12	1.21E-12	2.21E-06	1.21E-12	1.21E-12	1.21E-12	1.21E-12	NaN	1.21E-12
F5	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.99E-04	3.02E-11
F6	3.02E-11	3.69E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	4.62E-10	3.02E-11	3.02E-11
F7	3.02E-11	1.03E-06	1.03E-03	1.20E-08	3.02E-11	3.02E-11	3.02E-11	4.02E-01	1.06E-03
F8	3.02E-11	6.53E-07	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
F9	1.21E-12	8.14E-02	NaN	1.21E-12	1.21E-12	1.21E-12	1.08E-12	NaN	2.93E-05
F10	1.21E-12	3.08E-10	4.18E-02	1.21E-12	1.21E-12	1.21E-12	1.18E-12	NaN	2.71E-14
F11	1.21E-12	NaN	NaN	1.21E-12	1.21E-12	1.21E-12	6.62E-04	NaN	4.19E-02
F12	8.99E-11	1.41E-09	3.02E-11	3.02E-11	3.02E-11	3.02E-11	6.07E-11	3.02E-11	3.02E-11
F13	3.02E-11	4.08E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	4.08E-11	6.53E-08	3.02E-11
F14	8.30E-01	7.84E-01	1.45E-01	3.95E-01	7.62E-03	5.26E-04	4.73E-01	7.77E-09	8.15E-11
F15	3.02E-11	7.60E-07	4.98E-11	3.02E-11	1.61E-10	9.76E-10	1.91E-01	3.02E-11	7.06E-07
F16	2.36E-12	4.23E-03	3.02E-11	3.02E-11	1.75E-05	3.02E-11	7.96E-03	3.02E-11	3.34E-11
F17	1.21E-12	7.09E-08	3.02E-11	3.02E-11	3.02E-11	3.02E-11	4.08E-11	3.02E-11	3.02E-11
F18	2.97E-11	6.52E-08	4.69E-08	2.87E-10	4.21E-02	6.12E-10	6.12E-10	2.37E-10	3.47E-10
F19	2.16E-11	1.46E-10	3.02E-11	3.02E-11	9.76E-10	3.02E-11	4.20E-10	3.02E-11	3.02E-11
F20	6.63E-01	7.60E-07	3.02E-11	3.02E-11	8.15E-11	3.02E-11	3.79E-01	3.02E-11	3.02E-11
F21	8.30E-01	6.05E-07	3.02E-11	3.02E-11	8.15E-05	3.02E-11	1.12E-01	3.02E-11	3.02E-11
F22	9.51E-06	1.07E-09	3.02E-11	3.02E-11	2.03E-09	3.02E-11	1.96E-01	3.02E-11	3.02E-11
F23	6.53E-07	1.07E-07	3.02E-11	3.02E-11	1.33E-10	3.02E-11	1.02E-01	3.02E-11	3.02E-11

4 Engineering Problems

In this section, ABO is tested also with five constrained engineering design problems: Pressure vessel design, Rolling element bearing design, Tension/compression spring design, Cantilever beam design, and Gear train design. Mathematical models of these engineering design problems can be found in the corresponding literature. For the constraint problem, this paper uses the widely used penalty function method to deal with the constraints [22]. To ensure an equitable comparison, we maintain a consistent population size of 30 individuals across all optimization algorithms and perform 500 generations.

4.1 Pressure Vessel Design Problem

The pressure vessel design is an engineering problem initially presented by Kannan and Kramer [22]. As shown in Fig. 9, this design challenge involves the optimization of a pressure vessel's manufacturing cost, encompassing expenses related to materials, forming processes, and welding. The primary objective is to minimize the overall cost of producing the pressure vessel while ensuring it meets all design constraints and performance requirements.

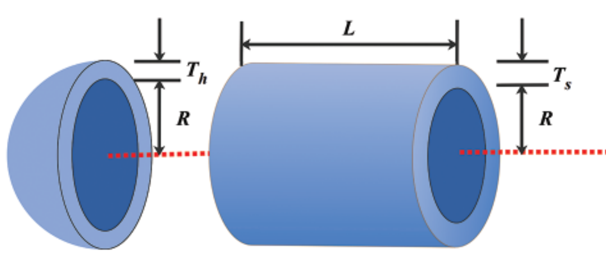

Figure 9: Pressure vessel design

We use ABO to optimize the problem and compare the experimental results with other algorithms. The results presented in Table 6 illustrate the statistical outcomes of the algorithms in terms of the optimal and variables. By utilizing ABO, it is possible to obtain the optimal function value of $f_1 = 6285.942$, with corresponding structure variables $x = (0.8140874, 0.4987349, 41.91206, 179.0077)$. The results show that ABO can find the optimal design of the pressure vessel problem with the lowest manufacturing cost compared with other algorithms.

Table 6: Comparison of optimum results for pressure vessel design

Algorithm	x_1	x_2	x_3	x_4	Optimal value
CHOA	1.25639	0.770405	64.3998	25.8942	9131.5308
ABO	0.8140874	0.4987349	41.91206	179.0077	**6285.942**
WOA	1.22615	0.660536	63.476	17.7501	7570.0419
GWO	1.25018	0.779287	58.5525	43.3575	8756.1309
SCA	0.8680526	0.6206409	43.22946	186.2539	7503.0508
TSA	1.22625	0.691567	63.2978	20.2611	7890.503
L-SHADE [23]	0.8525	0.5775	56.3105	65.7572	7672.4972
LSHADE-EpSin [23]	0.9330	0.6982	59.9952	47.5678	6854.5191
TEO [23]	2.5816	1.4787	47.1647	148.7692	32593.2941

4.2 Rolling Element Bearing Design

The design of rolling element bearings is a well-known engineering benchmark problem introduced by Rao and Tiwari in 2007. This problem aims to maximize the dynamic load-carrying capacity of rolling bearings, as illustrated in Fig. 10. The problem is framed by ten constraints and ten design variables.

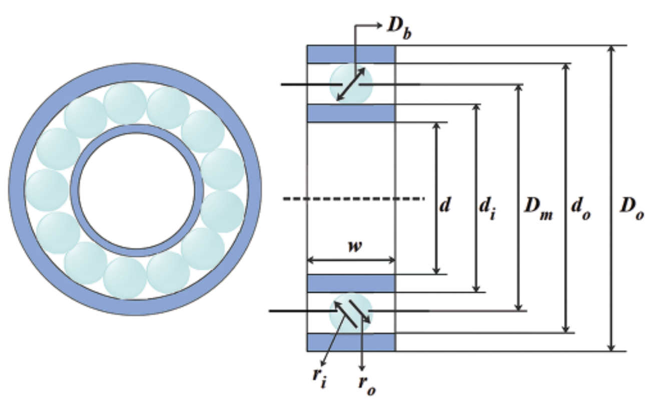

Figure 10: Rolling element bearing design

Table 7 shows the optimal values and optimal variables obtained by these algorithms on this problem. ABO can obtain the optimal function value $f_2 = -85380.5266$ with the structure variables $x = (125.6417, 21.40225, 11, 0.515001, 0.5154961, 0.4021139, 0.6181131, 0.3026727, 0.03466326, 0.6025443)$. The outcomes reveal that ABO can identify the optimal configuration for the design of rolling element bearings, maximizing their dynamic load-carrying capacity.

Table 7: Comparison of optimum results for rolling element bearing design

Algorithm	D_m	D_b	z	f_i	g_o	K_{Dmin}	K_{Dmax}	ε	e	ζ	Optimal value
CHOA	125	21.62214	10	0.515	0.6	0.5	0.7	0.3	0.02	0.651701	−81549.3006
ABO	125.6417	21.40225	11	0.515001	0.5154961	0.4021139	0.6181131	0.3026727	0.03466326	0.602544	−85380.5266
PSO	125	21	9	0.515	0.515	0.5	0.7	0.4	0.1	0.6	−35585.8093
WOA	125	21	9	0.515	0.515	0.4	0.6	0.3	0.053251	0.6	−72214.8943
GWO	125	21.09474	5	0.515	0.5693567	0.4246166	0.671365	0.325135	0.020982	0.610848	−49186.3383
Chaotic GWO [3]	125.68089	21.4174	11	0.5153	0.56367	0.49617	0.67858	0.3000001	0.05734287	0.616924	-81803. 645
SCA	125	20.82613	11	0.515	0.515	0.4	0.6	0.3	0.02	0.607731	−81343.9831
TSA	125	21.13977	11	0.515	0.5485597	0.4	0.627163	0.3	0.026647	0.6	−83519.8965
DE [22]	125.7192	21.2508	10.8654	0.5153	0.56021	0.4199	0.6197	0.3012	0.0472	0.6740	−83629.26366
PVS [27]	125.7191	21.4256	11	0.515	0.515	0.4004	0.6802	0.3	0.08	0.7	−81859.74121

4.3 Tension/Compression Spring Design Problem

The design problem introduced by Arora focuses on tension/compression spring design. This design problem, as depicted in Fig. 11, provides a visual representation of the optimization process for the spring design. The primary goal is to minimize the weight of the spring while ensuring it meets specific constraints, including minimum deflection, vibration frequency, and shear stress.

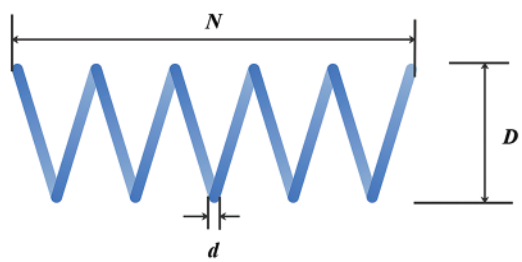

Figure 11: Tension spring design

The comparison results between ABO and other algorithms are given in Table 8. The superiority of ABO is evident as it attains the optimal solution with the variables set at $x = (0.0518975, 0.36175, 11)$, resulting in an objective function value of $f_3 = 0.012666$.

Table 8: Comparison of optimum results for tension/compression spring design

Algorithm	d	D	N	Optimal value
CHOA	0.05	0.310598	15	0.0132
ABO	0.0518975	0.36175	11	0.012666
PSO	0.064277	0.74196	3	0.015327
CPSO [22]	0.051728	0.357644	11	0.0126747
WOA	0.056439	0.4746	7	0.013606
GWO	0.057758	0.52053	6	0.013892
CGWO [3]	0.05169	0.356717	11	0.012665
CDE [3]	0.051609	0.354714	11	0.0126702

(Continued)

Table 8 (continued)

Algorithm	d	D	N	Optimal value
SCA	0.05	0.310549	15	0.013198
TSA	0.057676	0.51037	6	0.013582

4.4 Cantilever Beam Design

The cantilever beam design is given in Fig. 12. The design objective of the problem is to minimize the weight of the cantilever beam and the constraint is to satisfy a vertical displacement constraint [22]. It consists of five hollow members. Each unit is defined by distinct variables with a specific thickness, thus rendering this design problem to encompass five decision variables.

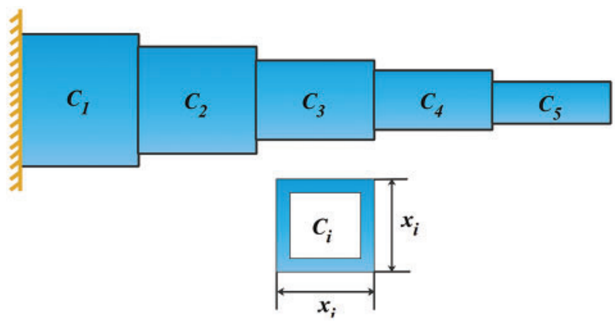

Figure 12: Cantilever beam design

Table 9 presents the optimized outcomes obtained from ABO as well as other competing methods. Notably, ABO achieves the optimal function value of f_4 =1.3422 by utilizing the structure variables $x = (5.8506, 5.4175, 4.4818, 3.6739, 2.0867)$. The conclusion that can be drawn is that ABO finds the optimal set of structural parameters for the cantilever beam design.

Table 9: Comparison of optimum results for cantilever beam design

Algorithm	x_1	x_2	x_3	x_4	x_5	Optimal value
CHOA	6.5777	5.4091	5.2894	3.4565	1.7035	1.4
ABO	5.8506	5.4175	4.4818	3.6739	2.0867	**1.3422**
WOA	8.0555	3.9969	4.6084	3.4613	2.0585	1.5281
GWO	6.0888	5.5787	4.3128	3.3401	2.2174	1.344
SCA	6.7659	4.833	4.9445	2.9159	3.5721	1.4372
TSA	6.0608	5.7167	4.3507	3.3217	2.1229	1.3461
AOA [23]	6.114257	5.352915	4.085839	3.181646	1.72922	1.3866
ACO [28]	5.0311	4.3266	2.0812	1.7568	3.3254	1.9311

4.5 Gear Train Design

The last engineering design problem is the gear train design [11]. As shown in Fig. 13, this problem has four integer decision variables, where T_a, T_b, T_d, and T_f represent the number of teeth of four different gears, respectively. Its objective is to minimize the transmission ratio cost of the gear train. The transmission ratio is given as $T_b/T_a \cdot T_d/T_f$.

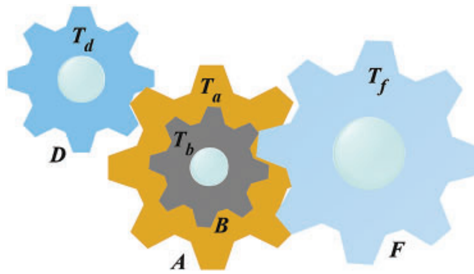

Figure 13: Gear train design

The comparison results obtained by ABO and other optimizers are given in Table 10. The optimal gear parameters obtained by the ABO optimization for this problem are $x = (34, 20, 13, 53)$ and the transmission ratio is $f_5 = 2.3078\text{E}{-}11$. The results demonstrate that ABO outperforms other algorithms in identifying the optimal design for the gear train, effectively minimizing the cost of transmission ratio.

Table 10: Comparison of optimum results for gear train design

Algorithm	T_a	T_b	T_d	T_f	Optimal cost
ARO	55	17	21	45	1.36E-09
CHOA	60	25	18	52	2.36E-09
ABO	34	20	13	53	**2.3078E-11**
PSO	60	13	40	60	2.73E-08
WOA	60	34	14	55	1.36E-09
GWO	59	14	28	46	2.46E-08
SCA	39	23	12	40	2.18E-08
TSA	26	12	15	48	2.36E-09
DE [22]	47	19	15	42	5.5209E-09

5 Conclusion

In this paper, a new metaheuristic optimization algorithm based on biological population, named ABO, is proposed. ABO mimics the foraging, jousting, mating, and eliminating behaviors of the African bison herd while introducing the parameters S and Q to balance the exploration and exploitation phases. To test the ability of ABO in global exploration and local exploitation, it is tested on 23 benchmark test functions and five constrained real-world engineering problems.

The experimental results of ABO demonstrate its superiority and the ability to solve real-world engineering problems, especially when compared to other existing algorithms. In future work, ABO could be further developed and modified to explore other aspects. One direction is to combine ABO with other algorithms to extend it to more fields, such as neural networks, image processing, etc. Another direction is to expand ABO into a multi-objective optimization tool. Incorporating multi-objective optimization capabilities could greatly enhance the utility of ABO, making it well-suited for solving complex problems with multiple objectives.

Acknowledgement: None.

Funding Statement: This work was supported in part by the National Natural Science Foundation of China (Grant No. U1731128); the Natural Science Foundation of Liaoning Province (Grant No. 2019-MS-174); the Foundation of Liaoning Province Education Administration (Grant No. LJKZ0279); the Team of Artificial Intelligence Theory and Application for the financial support.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Jian Zhao, Kang Wang; data collection: Jian Zhao; analysis and interpretation of results: Jian Zhao, Kang Wang, Jiacun Wang; draft manuscript preparation: Kang Wang, Jian Zhao, Xiwang Guo, Liang Qi. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. C. Fu, "Optimization via simulation: A review," *Ann. Oper. Res.*, vol. 53, pp. 199–247, 1994. doi: 10.1007/BF02136830.
[2] X. Cui, X. Guo, M. Zhou, J. Wang, S. Qin and L. Qi, "Discrete whale optimization algorithm for disassembly line balancing with carbon emission constraint," *IEEE Robot. Autom. Lett.*, vol. 8, no. 5, pp. 3055–3061, 2023. doi: 10.1109/LRA.2023.3241752.
[3] C. Lu, L. Gao, and X. Li, "Chaotic-based grey wolf optimizer for numerical and engineering optimization problems," *Memetic Comp.*, vol. 12, pp. 371–398, 2020. doi: 10.1007/s12293-020-00313-6.
[4] C. Lu, L. Gao, and J. Yi, "Grey wolf optimizer with cellular topological structure," *Expert. Syst. Appl.*, vol. 107, pp. 89–114, 2018. doi: 10.1016/j.eswa.2018.04.012.
[5] J. H. Holland, "Genetic algorithms," *Sci. Am.*, vol. 267, no. 1, pp. 66–73, 1992. doi: 10.1038/scientificamerican0792-66.
[6] R. Storn and K. Price, "Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces," *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, 1997. doi: 10.1023/A:1008202821328.
[7] D. Corne and M. A. Lones, *Evolutionary Algorithms*. Cham: Springer International Publishing, 2018, pp. 409–430.
[8] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl. Based Syst.*, vol. 96, pp. 120–133, 2016. doi: 10.1016/j.knosys.2015.12.022.
[9] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, 2009. doi: 10.1016/j.ins.2009.03.004.
[10] M. -H. Tayarani-N and M. -R. Akbarzadeh-T, "Magnetic-inspired optimization algorithms: Operators and structures," *Swarm Evol. Comput.*, vol. 19, pp. 82–101, 2014. doi: 10.1016/j.swevo.2014.06.004.

[11] R. V. Rao, V. J. Savsani, and D. Vakharia, "Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems," *Comput.-Aided Design*, vol. 43, no. 3, pp. 303–315, 2011. doi: 10.1016/j.cad.2010.12.015.

[12] E. K. Mousavirad, "Human mental search: A new population-based metaheuristic optimization algorithm," *Appl. Intell.*, vol. 47, no. 4, pp. 850–887, 2017. doi: 10.1007/s10489-017-0903-6.

[13] N. Satapathy, "Social group optimization (SGO): A new population evolutionary optimization technique," *Complex Intell. Syst.*, vol. 2, no. 4, pp. 173–203, 2016. doi: 10.1007/s40747-016-0022-8.

[14] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intell.*, vol. 1, no. 1, pp. 33–57, 2007. doi: 10.1007/s11721-007-0002-0.

[15] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, 2016. doi: 10.1016/j.advengsoft.2016.01.008.

[16] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, 2014. doi: 10.1016/j.advengsoft.2013.12.007.

[17] X. S. Yang, *Flower Pollination Algorithm for Global Optimization*. Berlin, Heidelberg: Springer, 2012, pp. 240–249.

[18] M. Khishe and M. Mosavi, "Chimp optimization algorithm," *Expert. Syst. Appl.*, vol. 149, pp. 113–338, 2020. doi: 10.1016/j.eswa.2020.113338.

[19] G. Dhiman, M. Garg, A. Nagar, V. Kumar, and M. Dehghani, "A novel algorithm for global optimization: Rat swarm optimizer," *J. Ambient Intell. Humaniz. Comput.*, vol. 12, no. 8, pp. 8457–8482, 2021. doi: 10.1007/s12652-020-02580-0.

[20] S. Kaur, L. K. Awasthi, A. Sangal, and G. Dhiman, "Tunicate swarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization," *Eng. Appl. Artif. Intell.*, vol. 90, pp. 103–541, 2020. doi: 10.1016/j.engappai.2020.103541.

[21] L. Abualigah, M. A. Elaziz, P. Sumari, Z. W. Geem, and A. H. Gandomi, "Reptile search algorithm (rsa): A nature-inspired meta-heuristic optimizer," *Expert. Syst. Appl.*, vol. 191, pp. 116–158, 2022. doi: 10.1016/j.eswa.2021.116158.

[22] L. Wang, Q. Cao, Z. Zhang, S. Mirjalili, and W. Zhao, "Artificial rabbits optimization: A new bio-inspired algorithm for solving engineering optimization problems," *Eng. Appl. Artif. Intell.*, vol. 114, 2022, Art. no. 105082. doi: 10.1016/j.engappai.2022.105082.

[23] F. A. Hashim and A. G. Hussien, "Snake optimizer: A novel meta-heuristic optimization algorithm," *Knowl. Based Syst.*, vol. 242, pp. 108–320, 2022. doi: 10.1016/j.knosys.2022.108320.

[24] J. B. Odili, M. N. M. Kahar, and S. Anwar, "African buffalo optimization: A swarm-intelligence technique," *Procedia Comput. Sci.*, vol. 76, pp. 443–448, 2015. doi: 10.1016/j.procs.2015.12.291.

[25] X. S. Yang, *A New Metaheuristic Bat-Inspired Algorithm*. Berlin, Heidelberg: Springer, pp. 65–74, 2010.

[26] F. Wilcoxon, *Individual Comparisons by Ranking Methods*. New York: Springer, 1992, pp. 196–202.

[27] J. P. Savsani and V. Savsani, "Passing vehicle search (PVS): A novel metaheuristic algorithm," *Appl. Math. Model.*, vol. 40, no. 5, pp. 3951–3978, 2016. doi: 10.1016/j.apm.2015.10.040.

[28] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, 2006. doi: 10.1109/MCI.2006.329691.