Sakarya University Journal of Science

ISSN : 2147-835X
Publisher : Sakarya University

Vol. 28, No. 3, 610-633, 2024
DOI: https://doi.org/10.16984/saufenbilder.1399655

Research Article

A Detailed Comparison of Two New Heuristic Algorithms Based on Gazelles Behavior

Emine Baş

Konya Technical University, Faculty of Engineering and Nature Sciences, Department of Software Engineering, Konya, Türkiye, ebas@ktun.edu.tr

ARTICLE INFO

Keywords:
Gazelle
Exploration
Exploitation
Benchmarks

Article History:
Received: 03.12.2023
Accepted: 29.04.2024
Online Available: 15.06.2024

ABSTRACT

In this study, Mountain Gazelle Optimization (MGO) and Gazelle Optimization Algorithm (GOA) algorithms, which have been newly proposed in recent years, were examined. Although MGO and GOA are different heuristic algorithms, they are often considered the same algorithms by researchers. This study was conducted to resolve this confusion and demonstrate the discovery and exploitation success of both algorithms. While MGO developed the exploration and exploitation ability by being inspired by the behavior of gazelles living in different groups, GOA model was developed by being inspired by the behavior of gazelles in escaping from predators, reaching safe environments and grazing in safe environments. MGO and GOA were tested on 13 classical benchmark functions in seven different dimensions and their success was compared. According to the results, MGO is more successful than GOA in all dimensions. GOA, on the other hand, works faster than MGO. Additionally, MGO and GOA were tested on three different engineering design problems. While MGO was more successful in the tension/compression spring design problem and welded beam design problems, GOA achieved better results in the pressure vessel design problem. The results show that MGO improves the ability to explore and avoid local traps better than GOA. MGO and GOA are also compared with three different heuristic algorithms selected from the literature (GSO, COA, and ZOA). According to the results, MGO has shown that it can compete with new algorithms in the literature. GOA, on the other hand, lags behind comparison algorithms.

1. Introduction

While an optimization problem describes a problem that has more than one feasible solution, optimization is the process of finding the best solution among all available solutions [1]. Optimization problems consist of decision variables, constraints and objective function [2]. Various methods, such as gradient-based methods and numerical calculations, have been proposed to solve optimization problems [1]. In an optimization problem, simply finding a solution is not enough. The cost and time involved in reaching this solution are important. Real-world problems often have multiple decision variables and complex nonlinear relationships. The inability of analytical methods to solve optimization problems has led to the emergence of meta-heuristic algorithms. Meta-heuristic algorithms are stochastic methods inspired by nature and its mechanisms, which try to send the initial population to the global optimum and provide appropriate solutions close to the global optimum in a reasonable time [3].

Many new metaheuristic algorithms have been proposed in recent years due to their success in solving many real-world problems. The most important of these are Mountain Gazelle Optimization (MGO) [4], Gazelle Optimization Algorithm (GOA) [5], Aquila Optimizer (AO) [1], Harris Hawks Optimization (HHO) [6], Slime Mould Algorithm (SMA) [7], Tunicate Swarm Algorithm (TSA) [8], Equilibrium Optimizer (EO) [9], SCA, Arithmetic Optimization Algorithm (AOA) [10], Marine

Cite as: E. Baş (2024). A Detailed Comparison of Two New Heuristic Algorithms Based on Gazelles Behavior, *Sakarya University Journal of Science*, 28(3), 610-633. https://doi.org/10.16984/saufenbilder.1399655

Predators Algorithm (MPA) [11], Zebra Optimization Algorithm (ZOA) [12], Crayfsh Optimization Algorithm (COA) [13], Golden Search Optimization Algorithm (GSO) [14], etc.

In recent years, many heuristic algorithms have been proposed for continuous optimization in the literature. This may also cause the similarities in the names of these algorithms. Similarities in the names of algorithms often cause confusion among researchers. Sometimes they are used interchangeably and sometimes cause pronunciation errors. For this reason, heuristic algorithms, which are often confused, are compared one to one in the literature. For example, Baş and Ülker proposed a comparison between SSA and SSO algorithm inspired in the behavior of the social spider for constrained optimization in 2021 [15]. SSA and SSO heuristic optimization algorithms are two different algorithms inspired by spider behavior.

The frequent use of SSO instead of SSA in the literature has pushed the SSA algorithm into the background and delayed its discovery by researchers. At the same time, in many studies, SSA was often abbreviated as SSO and sometimes SSO was abbreviated as SSA. This situation has caused even more complexity among algorithms with similar names. Baş and Ülker noticed this situation and introduced this study to the literature. Baş and Ihsan compared the Gray wolf and Kirill herd algorithms, which are two herd-based heuristic algorithms that have been widely preferred in recent years [16]. Their aim is to compare the success of both algorithms on CEC-C06 2019 functions in small and large sizes. In addition, they compared their success on big data optimization problems and drew attention to their success in large size problems.

In this study, two newly proposed heuristic algorithms, MGO and GOA, were examined in recent years. The reason why these heuristic algorithms were chosen is that the living groups inspired by both algorithms are similar. Both heuristic algorithms were inspired by the imitation of the social lifestyles of gazelle groups existing in nature. Although MGO and GOA are often considered the same heuristic algorithms in the literature, the discovery and exploitation capabilities of both algorithms differ. Four main factors in the life of mountain gazelles are used in the MGO mathematical model. These are single male herds, natal herds, solitary, territorial males and migrate in search of food [4]. MGO realizes its exploration and exploitation abilities with these four groups of mountain gazelles. The GOA model was inspired by the behavior of gazelles to escape from predators, reach safe environments and graze in safe environments. While the grazing behavior of gazelles in safe environments was used for the exploitation ability of the GOA, the behavior of escaping from predators was used for the exploration ability of the GOA.

When the literature was examined, it was noticed that a one-to-one comparison of these two similar algorithms was not given. In this study, these two algorithms are examined in detail and their success is shown in 13 classical benchmark functions in sizes 10, 20, 30, 50, 100, 500, and 1000. Both algorithms were compared according to best, worst, average, standard deviation, and time. The discovery and exploitation abilities of the algorithms are compared with each other. Wilcoxon signed rank test was performed on the results to determine whether there were semantic differences between MGO and GOA results. MGO and GOA were then compared on three different engineering design problems. According to the results, MGO explores the search space better than GOA and is less likely to get caught in local traps. MGO and GOA were compared with three different heuristic algorithms in recent years (GSO, COA, and ZOA) and it was examined whether they fell behind the literature.

The rest of the paper follows: In Section 2, MGO and GOA are explained in detail. Additionally, classical benchmarks used in comparisons are shown. In Section 3, MGO and GOA are compared on classical benchmarks in three low dimensions and four high dimensions. Statistical tests were performed on the results. In Section 4, MGO and GOA comparison results are interpreted and discussed.

2. General Methods

2.1. Mountain gazelle optimization algorithm (MGO)

Mountain gazelle is one of the gazelle species. It was a source of inspiration in the mathematical formation of MGO. Mountain gazelles are social creatures that live territorially and run very fast. Mountain gazelle territories consist of three groups. These are the mother-offspring herds, young male herds, and single males' territory [17]. Young male gazelles are in a constant struggle over the environment.

Mathematical model of MGO:

MGO is an optimization algorithm based on the social behavior and lifestyle of mountain gazelles. While modeling the MGO algorithm mathematically, basic concepts related to the social and group life of mountain gazelles are used. Four main factors in the life of mountain gazelles are used in the MGO mathematical model. These are single male herds, natal herds, solitary, territorial males, and migrate in search of food [4]. In MGO, each gazelle must be a member of one of the maternity herds, bachelor male herds, or solitary, territorial males. A new gazelle may also be born from these herds. In MGO, the best individuals are adult male gazelles. In MGO, candidate solutions added to the population are considered as gazelles in natal herds. In order to maintain the population number in each repetition, strong gazelles, that is, gazelles with quality solutions, remain in the population, while sick and old gazelles, that is, gazelles with poor quality solutions, are removed from the population. Thus, the herd population number is maintained [4].

In MGO, exploration and exploitation are carried out in four parallel mechanisms.

a- *Territorial Solitary Males (TSM):*

Male mountain gazelles are highly territorial. When they reach adulthood, that is, when they become strong enough, they create a territory. Regions are separated by large distances. Adult male gazelles fight for territory or possession of females. While young male gazelles try to occupy the territory or the female, adult males try to protect their environment. Adult male individuals are shown in Equations 1-5 [4].

$$TSM = male_{gazelle} - |(ri_1 \times BH - ri_2 \times X(t)) \times F| \times Cof_r \tag{1}$$

$$BH = X_{ra} \times \lfloor r_1 \rfloor + M_{pr} \times \lceil r_2 \rceil, \quad ra = \left\{ \left\lceil \frac{N}{3} \right\rceil \dots N \right\} \tag{2}$$

$$F = N_1(D) \times \exp\left(2 - Iter \times \left(\frac{2}{Iter_{max}}\right)\right) \tag{3}$$

$$Cof_i = \begin{cases} (a+1) + r_3, \\ a \times N_2(D), \\ r_4(D), \\ N_3(D) \times N_4(D)^2 \times cos\big((r_4 \times 2) \times N_3(D)\big), \end{cases} \tag{4}$$

$$a = -1 + Iter \times \left(\frac{-1}{Iter_{max}}\right) \tag{5}$$

where $male_{gazelle}$ is the position of the best global gazelle (adult male). ri_1 and ri_2 are random integers 1 or 2. BH is the young male herd coefficient vector. Cof_r is a randomly selected coefficient vector updated in each iteration. $X(t)$ is the position of the gazelle in the t^{th} iteration. X_{ra} is a random gazelle (young male). M_{pr} is the average number of search gazelles. They are chosen randomly. N is population size and r_1, r_2, r_3, and r_4 are random values between 0 and 1. D is the problem dimension. exp is exponential function. $Iter_{max}$ is the number of the maximum iteration and $Iter$ is the number of the current iteration. N_1 is a random number from the standard distribution. $N_2(D)$, $N_3(D)$, and $N_4(D)$ are random numbers in the normal range and the dimensions of the problem [4].

a- *Maternity Herds (MH):*

Maternity herds have a significant impact on the life of mountain gazelles. Maternity herds can give birth to male gazelles. The formation of maternal herds is shown in Equation 6 [4].

$$MH = \left(BH + Cof_{2,r}\right) + \left(ri_3 \times male_{gazelle} - ri_4 \times X_{rand}\right) \times Cof_{3,r} \qquad (6)$$

where BH is the young male herd coefficient vector. $Cof_{2,r}$ and $Cof_{3,r}$ are a randomly selected coefficient vectors. ri_3 and ri_4 are random integers 1 or 2. $male_{gazelle}$ is the position of the best global gazelle (adult male). X_{rand} is s the vector position of a random gazelle in the gazelle population [4].

b- Bachelor Male Herds (BMH):

As male gazelles mature, they tend to establish territories. They also want to capture female gazelles. This situation is shown in Equations 7-8.

$$BMH = (X(t) - D) + \left(ri_5 \times male_{gazelle} - ri_6 \times BH\right) \times Cof_r \qquad (7)$$

$$D = \left(|X(t)| + |male_{gazelle}|\right) \times (2 \times r_6 - 1) \qquad (8)$$

where $X(t)$ is the position of the gazelle in the t^{th} iteration. ri_5 and ri_6 are random integers 1 or 2. $male_{gazelle}$ is the position of the best global gazelle (adult male). BH is the young male herd coefficient vector. Cof_r are a randomly selected coefficient vectors. r_6 is also a random number between 0 and 1[4].

c- Migration to Search for Food (MSF):

Mountain gazelles constantly search for food sources in the search space, this situation is formulated by Equation 9 [4].

$$MSF = \left(Boundary_{upper} - Boundary_{lower}\right) \times r_7 + Boundary_{lower} \qquad (9)$$

where $Boundary_{upper}$ and $Boundary_{lower}$ are the upper and lower bounds of the problem. r_7 is a random number between 0 and 1. Figure 1 shows the pseudo-code of MGO and Figure 2 shows the flowchart of MGO [4].

Algorithm 1: Pseudo-code of MGO

Start:
Set MGO's Parameters (*N*=population size, $Iter_{max}$= maximum iteration, *D*=problem dimension)
Create a random gazelle population (X_i) (*i*=1, 2, 3, …, *N*)
Calculate objective values of the gazelle population
While (t<=$Iter_{max}$) **do**
    For (int *i*=1; *i*<=*N*; *i*++)
        Calculate *TSM* using Equation 1.
        Calculate *MH* using Equation 6.
        Calculate *BMH* using Equation 7.
        Calculate *MSF* using Equation 9.
        Calculate objective values of the *MSF*, *BMH*, *MH*, and *TSM*.
        Add *MSF*, *BMH*, *MH*, and *TSM* in gazelle population.
    For end
Gazelle population sorting in ascending order.
Update best gazelle.
Remove the four worst gazelles from the gazelle population.
End while
Return Global Gazelle
Stop.

Figure 1. The pseudo-code of MGO [4].

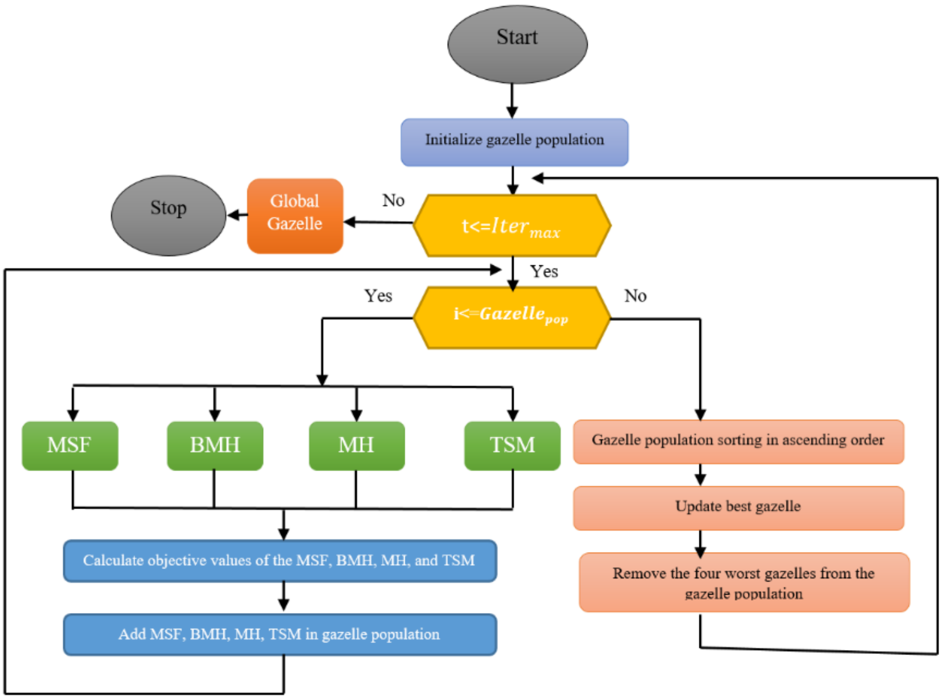


Figure 2. The flowchart of MGO [4]

.

2.2.Gazelle optimization algorithm (GOA)

Gazelles are creatures that can live in many areas, including arid areas and deserts. There are approximately 19 different species of gazelles worldwide [18]. They are mostly prey for other predators. Gazelles are light and fast and have a strong sense of hearing, sight and smell. These distinctive features are what allow them to escape predators. Gazelles are herbivores. They mostly socialize by living in groups. This situation is usually due to security. The more members there are in the group, the more secure the herd is. Sometimes groups of gazelles can group together depending on gender. Gazelles give birth once or twice a year. Reproduction generally occurs in seasons when water and food resources are abundant [5]. GOA was modeled on some characteristics of gazelles. These features can be listed as follows:

- The most notable aspects are grazing and escaping from predators.
- The grazing feature of gazelles can be used for exploitation. This should occur when there are no predators around.
- The situation of gazelles escaping from predators and reaching a safe environment has been used for exploration.

In GOA, initially all gazelles are randomly positioned in the search space. This situation is shown in Equations 10-11. Here *D* indicates the problem size (D=1, 2, …, d) and *N* indicates the population size (N=1, 2, …, n). *X* represents the gazelle population (X=1, 2, …, N). The X matrix is created at the lower and upper limits where the problem is defined. $Boundary_{upper}$ and $Boundary_{lower}$ are the upper and lower bounds of the problem. $rand$ is a random number [5].

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d-1} & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d-1} & x_{2,d} \\ . & . & . & . & . \\ . & . & x_{i,j} & . & . \\ . & . & . & . & . \\ x_{n,1} & x_{n,2} & \cdots & x_{n,d-1} & x_{n,d} \end{bmatrix} \quad (10)$$

$$x_{i,j} = rand \times \left(Boundary_{upper,j} - Boundary_{lower,j}\right) + Boundary_{lower,j} \quad (11)$$

In each iteration, search agents produce a solution in a candidate. The $Elite$ matrix is formed as best solution so far. This matrix is used in the displacement equations of gazelles in later stages. The $Elite$ matrix is shown in Equation 12 [5]. The elite will be updated at the end of each iteration if a better gazelle replaces the best gazelle. $x'_{i,j}$ represents the position vector of the top gazelle [5].

$$Elite = \begin{bmatrix} x'_{1,1} & x'_{1,2} & \dots & x'_{1,d-1} & x'_{1,d} \\ x'_{2,1} & x'_{2,2} & \dots & x'_{2,d-1} & x'_{2,d} \\ . & . & . & . & . \\ . & . & x'_{i,j} & . & . \\ . & . & . & . & . \\ x'_{n,1} & x'_{n,2} & \dots & 'x_{n,d-1} & x'_{n,d} \end{bmatrix} \quad (12)$$

The Brownian motion:

Equation 13 shows the standard brownian motion at point x (μ = 0 and σ^2 = 1) [19].

$$fB(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) = \frac{1}{\sqrt{2\pi}} exp\left(-\frac{x^2}{2}\right) \quad (13)$$

The Le´vy flight:

A random walk is performed using le'vy flight. Le'vy flight is shown in Equations 14-17 [20].

$$L(x_j) \approx |x_j|^{1-\alpha} \quad (14)$$

where α represents power-law exponent (α=(1,2]). x_j represents the flight distance [20].

$$fL(x;\alpha,\gamma) = \frac{1}{\pi}\int_0^{\infty} exp(-\gamma q^{\alpha})\, cos(qx)\delta q \quad (15)$$

where γ represents the scale unit [20].

$$Levy(\alpha) = 0.05 \times \frac{x}{|y|^{\frac{1}{\alpha}}} \quad (16)$$

where $x = Normal(0,\sigma_x^2)$ and $y = Normal(0,\sigma_y^2)$ [20].

$$\sigma_x = \left[\frac{\Gamma(1+\alpha)\sin\left(\frac{\pi\alpha}{2}\right)}{\Gamma\left(\frac{(1+\alpha)}{2}\right)\alpha 2^{\frac{(\alpha-1)}{2}}}\right]^{1/\alpha}, \sigma_y = 1,\ and\ \alpha = 1 \quad (17)$$

Exploitation:

In the exploitation phase of GOA, it is assumed that gazelles graze without predators or that the hunter follows the gazelles. At this stage, brownian motion was used in gazelle walks. This movement is shown in Equation 18 [5].

$$X_{i+1} = X_i + s.R *.R_B *.(Elite_i - R_B *.X_i) \quad (18)$$

where X_{i+1} shows the position of the gazelle at the next iteration. X_i shows the position of the gazelle at the current iteration. s shows grazing speed of the gazelles. R is a vector of uniform random numbers [0,1]. R_B shows a vector containing random numbers representing the Brownian motion [5].

Exploration:

The exploration phase begins when gazelles see a predator. Gazelles start running and hunters chase them. Levy flight was used at this stage. Gazelles can make sudden changes in direction. With each iteration, a direction change was made in the GOA. The mathematical model of the gazelle's behavior when it detects the predator is as shown in Equation 19 [5].

$$X_{i+1} = X_i + S.\mu.R *.R_L *.(Elite_i - R_L *.X_i) \quad (19)$$

where S shows the top speed. R_L shows a vector of random numbers based on Le´vy distributions. The movement of a gazelle being chased by a predator is shown in Equations 20-21 [5].

$$X_{i+1} = X_i + S.\mu.R *.CF *.R_B *.(Elite_i - R_L *.X_i) \quad (20)$$

$$CF = \left(1 - \frac{iter}{Iter_{max}}\right)^{\left(2\times\frac{iter}{Iter_{max}}\right)} \quad (21)$$

where CF shows the cumulative effect of the predator [5].

The PSR value expresses the success rate of the predator. It affects the gazelle's ability to escape, which means the algorithm avoids getting stuck in the local minimum. The PSR effect on the algorithm is shown in Equations 22-23 [5]. Figure 3 shows the pseudo-code of GOA and Figure 4 shows the flowchart of GOA [5].

$$X_{i+1} = \begin{cases} X_i + CF[B_{lower} + R *.(B_{upper} - B_{lower})] *.U & if\ r \le PSR \\ X_i + [PSR(1-r) + r](X_{r1} - X_{r2}) & else \end{cases} \quad (22)$$

$$U = \begin{cases} 0, & if\ r < 0.34 \\ 1, & otherwise \end{cases} \quad (23)$$

Algorithm 1: Pseudo-code of GOA

```
Start:
Set  GOA's Parameters (N=population size, Iter_max= maximum iteration, D=problem dimension)
Set  GOA's Parameters (s=[0,1]; μ = [−1,1] ; S=88kmph; PRS=0.34)
Set  GOA's Parameters (R and r random numbers [0,1])
Create a random gazelle population (X_i) (i=1, 2, 3, …, N)
While (t<=Iter_max) do
      Calculate objective values of the gazelle population.
      Create Elite gazelle matrix using Equation 12.
     If r<0.5 then
         Update gazelle using Equation 18.
     Else
         If mod(iter,2)==0  then
            μ=-1;
         Else
            μ=1;
         If end
         If iter<size(gazelle,1)/2  then
            Update gazelle using Equation 19.
         Else
            Update gazelle using Equation 20.
         If end
     End If
Update fitness values
Update Elite gazelle
Update best gazelle matrix
Applying PRS effect using  Equation 22.
End while
Return Global Gazelle
Stop.
```

Figure 3. The pseudo-code of GOA

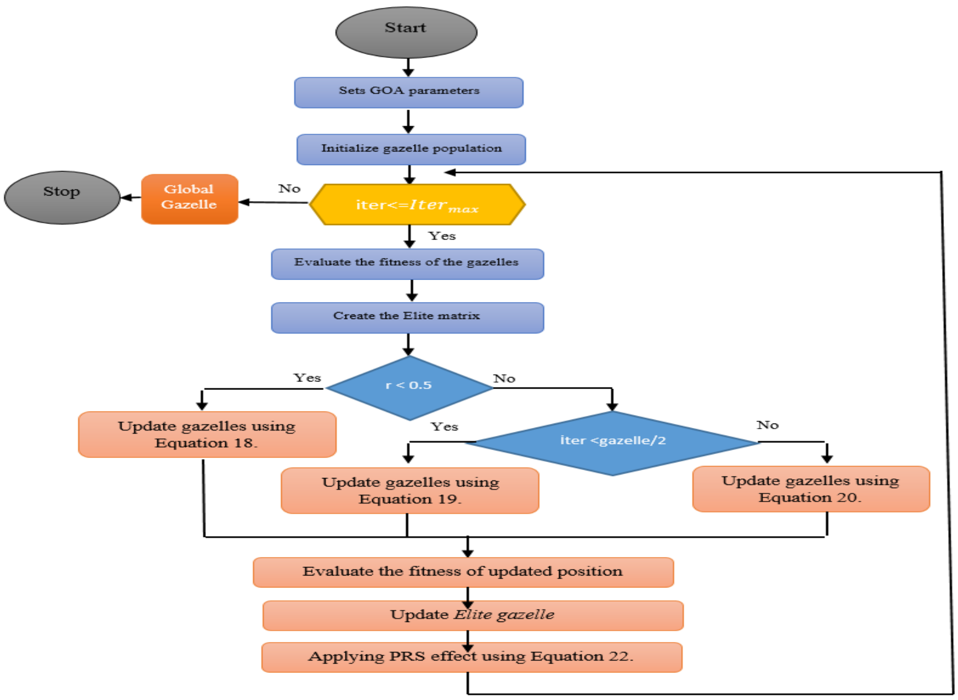


Figure 4. The flowchart of GOA

2.3. Classical benchmark functions

In this study, comparisons of MGO and GOA algorithms were compared on 13 classical test functions. These test functions are taken from https://www.sfu.ca/~ssurjano/optimization.html [1]. The classic test function consists of 7 unimodel and 6 multimodel test functions. Table 1 and Table 2 shows unimodel and multimodel test functions, respectively. Two different groups of benchmark functions were selected in this study. The reason for choosing these function groups is to test the performance of MGO and GOA from different aspects.

The first group of functions, single-mode benchmark functions, has a single minimum and thus the utilization and convergence rates of MGO and GOA are tested. The second group of functions, multimodal benchmark functions, have multiple minima, making them more challenging than single-mode benchmarks. Thus, the exploration and exploitation capabilities of MGO and GOA can be tested by multimodal benchmark functions. Figures 5-7 show 3D plots of F1 function, F9 function, and F11 function ([1, 21, 22]).

Table 1. The unimodal benchmark test functions of the mathematical formulations [1]

No	Range	F_{min}	Formulation
F1	$[-100,100]$	0	$f1(\vec{x}) = \sum_{i=1}^{D} x_i^2$
F2	[-10, 10]	0	$f2(\vec{x}) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$
F3	$[-100,100]$	0	$f3(\vec{x}) = \sum_{i=1}^{D} (\lfloor x_i + 0.5 \rfloor)^2$
F4	$[-100,100]$	0	$f4(\vec{x}) = \max_i \{\lvert x_i \rvert, 1 \leqslant i \leqslant D\}$
F5	$[-30,30]$	0	$f5(\vec{x}) = \sum_{i=1}^{D-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$
F6	$[-100,100]$	0	$f6(\vec{x}) = \sum_{i=1}^{D} ([x_i + 0.5])^2$
F7	$[-1.28,1.28]$	0	$f7(\vec{x}) = \sum_{i=1}^{D} \mathrm{i} x_i^4 + random[0,1)$

Table 2. The multimodal benchmark test functions of the mathematical formulations [1]

No.	Range	F_{min}	Formulation
F8	[-500, 500]	-418,9829×5	$f8(\vec{x}) = \sum_{i=1}^{n} - x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$
F9	[-5.12, 5.12]	0	$f9(\vec{x}) = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i) + 10]$
F10	$[-32,32]$	0	$f10(\vec{x}) = -20 \exp\left\{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{D} x_i^2}\right\} - \exp\left\{\frac{1}{D}\sum_{i=1}^{D} \cos(2\pi x_i)\right\} + 20 + e$
F11	[-600, 600]	0	$f11(\vec{x}) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
F12	$[-50,50]$	0	$f12(\vec{x}) = \frac{\pi}{D}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_D - 1)^2\right\} + \sum_{i=1}^{D} u(x_i, 10, 100, 4)$; $y_i = 1 + \frac{1}{4}(x_i + 1) u_{x_i,a,k,m} = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \leqslant x_i \leqslant a \\ k(x_i - a)^m & x_i < -a \end{cases}$
F13	$[-50,50]$	0	$f13(\vec{x}) = \frac{1}{10}\left\{\sin^2(\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2[1 + \sin^2(2\pi x_{i+1})]\right\} + \sum_{i=1}^{D} u(x_i, 5, 100, 4)$

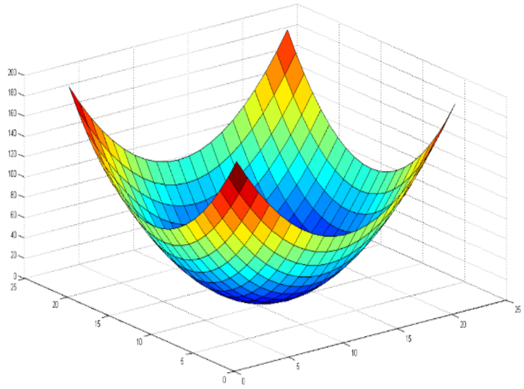


Figure 5. 3D plots of F1 function ([1], [21], [22])

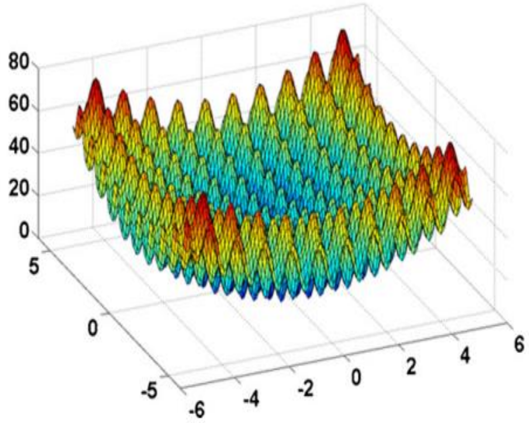


Figure 6. 3D plots of F9 function ([1], [21], [22])

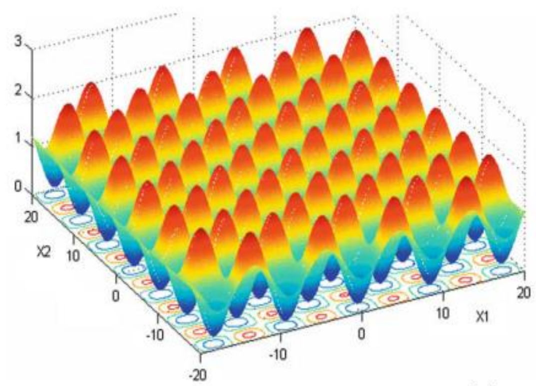


Figure 7. 3D plots of F11 function ([1], [21], [22])

3. Results and Discussion

3.1. The comparison of MGO and GOA on 13 classical benchmark functions

In this section, the results of MGO and GOA algorithms on 13 classical benchmarks in seven different dimensions are compared. The codes of the MGO and GOA algorithms were obtained from the mathworks library (https://ww2.mathworks.cn/en/). MGO and

GOA algorithms were run on a PC with Windows 10 Home with 64 bits operating system, Intel(R) Core(TM) i5 1.19 GHz CPU, and 12 GB RAM. Comparisons were made under equal conditions to ensure fairness. Parameter settings for both algorithms are shown in Table 3.

Table 3. The parameter settings for MGO and GOA

Parameters	Values
Population size (N)	30
Maximum iterations	200 ($Iter_{max}$)
Dimension (D)	10, 30, 50, 100, 500, 1000
GOA	PSRs = 0.34; S=0.88

There is no need for special fixed parameter tuning for MGO. The *PSRs* and *S* parameter settings used in GOA are determined as 0.34 and 0.88, respectively. These parameter settings were determined by the authors who proposed the GOA algorithm in the literature [5]. In this paper, *PSRs* and *S* parameters are used at similar values. Both algorithms were run independently 20 times on 13 classical benchmarks. Best, worst, average (Mean), standard deviation (Std), and average time (Time) statistical evaluations were made on the results obtained. The results are shown for seven different sizes in Tables 4-10. Wilcoxon signed rank test was performed on the results to determine whether there were semantic differences between the MGO and GOA results. *p* and *h* values are shown in Table 11. The Wilcoxon signed rank test is a statistical test used to compare two samples of data to detect significant differences between them [23]. In this test, *p* value or *h* value are the criteria that determine the superiority of one algorithm over another. If the *p* value is equal to or above 0.05, there is no semantic difference and the *h* value is 0. If the *p* value is below 0.05, there is a semantic difference and the *h* value is 1.

Table 4. The results of the MGO and GOA on classical benchmark functions (D=10)

F_ID	MGO					GOA				
	Best	***Worst***	***Mean***	***Std***	***Time***	***Best***	***Worst***	***Mean***	***Std***	***Time***
F1	2.88e-52	6.80e-46	7.01e-47	1.70e-46	0.727	2.0e-38	1.76e-17	8.85e-19	3.83e-18	0.511
F2	8.65e-30	1.29e-25	2.27e-26	3.98e-26	0.751	6.93e-23	2.81e-11	1.42e-12	6.12e-12	0.721
F3	2.25e-17	5.99e-09	3.46e-10	1.30e-09	1.365	7.870e-16	1.55E-04	8.03e-06	3.38e-05	0.950
F4	1.01e-20	1.36e-15	6.96e-17	2.97e-16	1.069	1.90e-14	1.05E-03	6.06e-05	2.29E-04	0.722
F5	1.75e-12	1.68e-04	8.44e-06	3.66e-05	1.292	4.00E+00	5.85 e+00	5.19 e+00	0.48 e+00	0.712
F6	7.01e-19	7.56e-14	9.51e-15	2.04e-14	1.335	8.84e-05	1.70 E-03	5.98E-04	3.89E-04	0.724
F7	2.64e-05	1.18e-03	5.09e-04	3.39e-04	1.281	6.01E-04	6.24E-03	2.52e-03	1.26 e-03	0.758
F8	-4.19e+03	-4.19e+03	-4.19e+03	7.37e-12	1.401	-4.00E+03	-3.14E+03	-3.60E+03	2.42E+02	0.751
F9	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.378	0.00e+00	2.02e+00	0.15e+00	0.48e+00	0.712
F10	8.88e-16	4.44e-15	1.78e-15	1.54e-15	1.384	4.44e-15	5.02e-09	2.53e-10	1.09e-09	0.750
F11	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.178	0.00e+00	8.45E-02	1.04E-02	2.27E-02	0.784
F12	6.14e-24	1.53e-18	1.51e-19	3.40e-19	1.530	5.84e-06	3.52E-04	1.26E-04	9.14E-05	0.955
F13	7.72e-26	2.45e-20	1.41e-21	5.30e-21	1.529	1.71E-04	3.20E-03	1.03E-03	7.89E-04	0.980

Table 5. The results of the MGO and GOA on classical benchmark functions (D=20)

F_ID	MGO					GOA				
	Best	***Worst***	***Mean***	***Std***	***Time***	***Best***	***Worst***	***Mean***	***Std***	***Time***
F1	1.93E-36	7.18E-29	4.62E-30	1.60E-29	0.760	1.40E-25	8.46E-10	4.84E-11	1.85E-10	0.481
F2	4.45E-22	2.95E-18	3.79E-19	7.41E-19	1.054	2.05E-16	1.45E-05	7.27E-07	3.16E-06	0.455
F3	1.78E-11	2.08E-04	4.76E-05	6.99E-05	2.097	1.26E-05	1.16E+02	6.49E+00	2.53E+01	0.720
F4	1.31E-13	1.11E-10	1.43E-11	2.76E-11	1.082	1.03E-08	3.35E-02	3.18E-03	9.38E-03	0.468
F5	2.27E-14	3.75E-09	2.33E-10	8.21E-10	1.178	1.54E+01	1.86E+01	1.67E+01	6.37E-01	0.474
F6	1.46E-08	3.21E-06	7.31E-07	1.03E-06	1.342	8.62E-03	2.24E-01	9.06E-02	6.09E-02	0.467
F7	7.41E-05	3.12E-03	8.36E-04	7.02E-04	1.363	8.68E-04	2.69E-02	6.76E-03	5.86E-03	0.562
F8	-8.38E+03	-8.38E+03	-8.38E+03	1.18E-06	1.271	-6.02E+03	-4.64E+03	-5.23E+03	3.94E+02	0.477
F9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.333	0.00E+00	1.52E+01	3.08E+00	5.23E+00	0.489
F10	8.88E-16	4.44E-15	3.73E-15	1.42E-15	1.515	1.15E-14	6.46E-03	3.37E-04	1.41E-03	0.480
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.435	0.00E+00	7.65E-02	8.14E-03	2.00E-02	0.495
F12	1.01E-18	2.60E-13	4.25E-14	6.03E-14	1.769	1.73E-03	1.54E-02	5.59E-03	3.22E-03	0.614
F13	8.01E-19	1.56E-15	1.65E-16	3.42E-16	1.622	1.18E-02	1.97E-01	7.63E-02	4.62E-02	0.642

Table 6. The results of the MGO and GOA on classical benchmark functions (D=30)

F_ID	MGO					GOA				
	Best	***Worst***	***Mean***	***Std***	***Time***	***Best***	***Worst***	***Mean***	***Std***	***Time***
F1	3.99e-37	1.52e-27	8.630e-29	3.31e-28	1.089	1.02E-22	2.94E-04	2.40E-05	7.4E-05	0.508
F2	5.43e-19	1.57e-16	2.38e-17	4.04e-17	1.424	5.49E-14	4.30E-07	2.17E-08	9.36E-08	0.713
F3	5.27e-07	2.84E-02	2.87E-03	6.64E-03	2.816	3.85E-02	7.13E+02	4.93E+01	1.55E+02	1.336
F4	3.25e-13	3.85e-10	9.14e-11	1.05e-10	1.290	8.45E-07	6.10E-01	5.02E-02	1.38E-01	0.750
F5	3.60e-16	8.37e-10	7.48e-11	1.94e-10	1.530	2.65E+01	2.91E+01	2.73E+01	6.35E-01	0.907
F6	6.10e-08	8.42e-05	2.03e-05	2.43e-05	1.372	3.29E-01	1.42E+00	7.91E-01	2.92E-01	0.822
F7	7.77e-05	2.10E-03	8.18E-04	5.89E-04	1.491	3.75E-03	1.95E-02	7.68E-03	3.96E-03	0.867
F8	-1.26E+04	-1.26E+04	-1.26E+04	2.32E-04	1.431	-8.64E+03	-5.23E+03	-6.84E+03	7.55E+02	0.814
F9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.542	0.00E+00	3.25E+01	2.66E+00	7.33E+00	0.827
F10	8.88e-16	4.71e-14	6.22e-15	9.50e-15	1.559	6.07E-12	5.12E-04	2.56E-05	1.11E-04	0.798
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.514	0.00E+00	2.65E-02	2.08E-03	6.50E-03	0.842
F12	3.52e-15	4.47e-12	6.00e-13	1.09e-12	2.017	1.24E-02	5.70E-02	3.51E-02	1.30E-02	1.082
F13	7.53e-17	4.6e-14	7.93e-15	1.20e-14	2.045	2.70E-01	9.37E-01	5.48E-01	1.68E-01	1.054

Table 7. The results of the MGO and GOA on classical benchmark functions (D=50)

F_ID	MGO					GOA				
	Best	***Worst***	***Mean***	***Std***	***Time***	***Best***	***Worst***	***Mean***	***Std***	***Time***
F1	1.11E-30	2.55E-23	1.35E-24	5.55E-24	1.212	1.54E-17	4.24E-10	3.91E-11	1.12E-10	0.560
F2	1.58E-18	3.13E-14	2.76E-15	6.75E-15	1.476	6.92E-12	2.57E-03	1.31E-04	5.59E-04	0.793
F3	2.17E-06	1.62E+00	8.25E-02	3.53E-01	3.603	3.37E+01	3.32E+03	8.27E+02	8.84E+02	1.755
F4	4.76E-13	4.27E-09	4.50E-10	9.97E-10	1.341	2.77E-05	6.47E-01	5.05E-02	1.55E-01	0.72
F5	1.17E-12	1.09E-08	1.04E-09	2.48E-09	1.390	4.66E+01	4.87E+01	4.81E+01	6.37E-01	0.938
F6	3.83E-07	5.49E-03	6.81E-04	1.21E-03	1.325	2.30E+00	4.88E+00	3.57E+00	6.65E-01	0.812
F7	7.99E-05	3.43E-03	1.10E-03	8.33E-04	1.608	3.02E-03	8.15E-02	1.55E-02	1.77E-02	0.998
F8	-2.09E+04	-2.09E+04	-2.09E+04	3.85E-03	1.38	-1.04E+04	-7.54E+03	-8.47E+03	8.69E+02	0.836
F9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.364	0.00E+00	8.82E+01	1.09E+01	2.63E+01	0.878
F10	8.88E-16	1.15E-13	1.40E-14	2.55E-14	1.381	2.47E-10	2.05E-02	1.11E-03	4.46E-03	0.888
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.387	0.00E+00	6.53E-01	3.56E-02	1.42E-01	0.928
F12	6.32E-14	3.19E-11	4.15E-12	7.80E-12	2.458	7.24E-02	2.07E-01	1.24E-01	3.15E-02	1.219
F13	1.52E-15	5.08E-13	9.82E-14	1.24E-13	2.316	1.69E+00	5.03E+00	2.61E+00	6.91E-01	1.229

Table 8. The results of the MGO and GOA on classical benchmark functions (D=100)

F_ID	MGO					GOA				
	Best	***Worst***	***Mean***	***Std***	***Time***	***Best***	***Worst***	***Mean***	***Std***	***Time***
F1	3.43E-31	1.98E-24	2.73E-25	5.13E-25	1.021	6.53E-14	8.38E+01	4.20E+00	1.83E+01	0.849
F2	7.48E-17	6.56E-14	9.33E-15	1.71E-14	0.961	7.11E-10	2.13E-01	1.06E-02	4.64E-02	1.085
F3	1.89E-05	5.07E+00	9.08E-01	1.58E+00	6.223	7.51E+02	2.17E+04	8.43E+03	6.36E+03	3.278
F4	1.71E-12	2.22E-08	2.49E-09	5.03E-09	1.761	6.17E-04	5.55E-01	6.17E-02	1.22E-01	0.657
F5	2.47E-12	2.56E-08	2.48E-09	5.89E-09	1.831	9.78E+01	6.09E+04	3.14E+03	1.32E+04	0.986
F6	1.13E-05	2.83E-02	3.67E-03	7.58E-03	1.651	1.10E+01	1.70E+01	1.39E+01	1.31E+00	1.281
F7	2.42E-04	4.25E-03	1.56E-03	1.08E-03	2.331	4.67E-03	6.14E-02	1.59E-02	1.54E-02	1.649
F8	-4.19E+04	-4.19E+04	-4.19E+04	1.83E-02	1.778	-2.28E+04	-1.02E+04	-1.22E+04	2.77E+03	1.365
F9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.591	2.50E-12	2.94E+01	1.72E+00	6.44E+00	1.338
F10	4.44E-15	7.61E-13	1.32E-13	2.18E-13	1.416	3.77E-08	4.27E+00	3.35E-01	1.05E+00	1.383
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.715	4.77E-14	1.37E+00	7.61E-02	2.98E-01	1.439
F12	2.60E-14	2.61E-11	3.51E-12	7.04E-12	2.764	2.54E-01	4.42E-01	3.37E-01	5.86E-02	2.230
F13	1.86E-14	2.31E-12	6.56E-13	6.61E-13	2.82	7.88E+00	1.03E+02	1.70E+01	2.27E+01	2.267

Table 9. The results of the MGO and GOA on classical benchmark functions (D=500)

F_ID	MGO					GOA				
	Best	***Worst***	***Mean***	***Std***	***Time***	***Best***	***Worst***	***Mean***	***Std***	***Time***
F1	1.88e-27	1.19e-19	7.00e-21	2.59e-20	2.892	1.79E-09	1.81E+00	9.09E-02	3.94E-01	1.737
F2	7.12e-17	2.80e-11	1.68e-12	6.06e-12	3.147	5.85E-07	5.70E+80	2.85E+79	1.24E+80	2.597
F3	5.54E-01	1.23E+04	1.56E+03	2.90E+03	33.232	6.04E+04	5.56E+05	3.10E+05	1.55E+05	18.250
F4	2.20e-11	9.30e-08	8.12e-09	2.01e-08	2.718	1.04E-01	4.08E+01	1.19E+01	1.40E+01	2.585
F5	5.75e-12	3.78e-07	4.70e-08	8.35e-08	2.555	4.98E+02	5.39E+05	3.05E+04	1.17E+05	2.605
F6	2.16e-05	3.18E-01	5.26E-02	8.94E-02	2.791	1.04E+02	5.78E+02	1.33E+02	1.02E+02	2.502
F7	9.76e-05	9.15E-03	2.35E-03	2.09E-03	5.476	5.87E-03	1.31E-01	3.37E-02	3.17E-02	4.294
F8	-2.09E+05	-2.09E+05	-2.09E+05	6.13E-01	3.149	-3.76E+04	-2.26E+04	-2.72E+04	3.91E+03	2.431
F9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.573	3.52E-10	3.95E+03	2.18E+02	8.62E+02	2.706
F10	1.87E-14	2.26E-11	2.75E-12	5.88E-12	2.887	3.65E-06	1.06E+01	1.13E+00	2.99E+00	2.795
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.767	8.25E-10	3.51E+02	1.75E+01	7.64E+01	2.805
F12	2.54E-15	3.18E-11	4.16E-12	9.29E-12	8.111	7.85E-01	3.36E+01	2.82E+00	7.16E+00	6.266
F13	9.36E-14	5.53E-10	4.74E-11	1.21E-10	9.073	4.91E+01	2.05E+02	5.84E+01	3.39E+01	5.880

Table 10. The results of the MGO and GOA on classical benchmark functions (D=1000)

F_ID	MGO					GOA				
	Best	***Worst***	***Mean***	***Std***	***Time***	***Best***	***Worst***	***Mean***	***Std***	***Time***
F1	4.04e-26	8.78e-17	4.42e-18	1.91e-17	4.525	1.40E-07	1.51E+05	1.24E+04	3.81E+04	1.860
F2	5.20e-14	1.09e-11	2.32e-12	2.820e-12	5.260	5.27E+02	2.30E+163	1.15E+162	null	1.925
F3	2.43E+00	6.37E+04	1.41E+04	1.83E+04	82.032	3.87E+05	3.54E+06	1.58E+06	8.21E+05	26.34
F4	2.82e-12	2.98e-08	5.11e-09	9.04e-09	4.585	6.67E-01	9.97E+01	3.18E+01	3.00E+01	1.858
F5	1.99e-09	2.90e-06	2.54e-07	6.18e-07	4.778	9.98E+02	8.80E+07	4.44E+06	1.92E+07	1.947
F6	3.70e-06	1.04E+00	1.89E-01	2.91E-01	4.669	2.28E+02	4.94E+02	2.47E+02	5.67E+01	1.885
F7	1.12E-04	8.41E-03	2.14E-03	2.29E-03	9.406	1.00E-02	2.05E+00	1.63E-01	4.42E-01	3.577
F8	-4.19E+05	-4.19E+05	-4.19E+05	3.70E-01	5.281	-4.12E+04	-3.23E+04	-3.61E+04	2.28E+03	2.181
F9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.977	6.52E-09	4.68E+02	2.80E+01	1.02E+02	2.029
F10	7.99E-15	2.72E-11	3.58E-12	6.75E-12	5.136	1.15E-05	8.73E+00	4.40E-01	1.90E+00	2.099
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	5.125	1.20E-08	4.90E+02	2.45E+01	1.07E+02	2.121
F12	1.03E-16	2.19E-11	4.24E-12	6.16E-12	14.966	8.99E-01	2.28E+01	2.11E+00	4.74E+00	5.634
F13	1.77E-12	2.01E-09	1.91E-10	4.41E-10	15.096	9.90E+01	2.26E+03	2.24E+02	4.69E+02	5.605

Table 11. The wilcoxon signed rank test results of the MGO and GOA on classical benchmark functions (D={10, 20, 30, 50, 100, 500, 1000}) (p-value) (h-value)

F_ID	***MGO-GOA***													
	D=10		***D=20***		***D=30***		***D=50***		***D=100***		***D=500***		***D=100***	
	p	***h***	***p***	***h***	***p***	***h***	***p***	***h***	***p***	***h***	***p***	***h***	***p***	***h***
F1	0.0002	1	0.00049	1	0.00049	1	0.00024	1	0.000244	1	0.000244	1	0.000244	1
F2	0.0049	1	0.00049	1	0.00049	1	0.00024	1	0.000244	1	0.000244	1	0.000244	1
F3	0.00097	1	0.00024	1	0.00024	1	0.00049	1	0.000244	1	0.000244	1	0.000244	1
F4	0.0048	1	0.00024	1	0.00024	1	0.00024	1	0.000244	1	0.000244	1	0.000244	1
F5	0.00098	1	0.00024	1	0.00024	1	0.00098	1	0.000244	1	0.000244	1	0.000244	1
F6	0.00024	1	0.00049	1	0.00049	1	0.00024	1	0.000244	1	0.000244	1	0.000244	1
F7	0.0068	1	0.00024	1	0.00024	1	0.00024	1	0.000244	1	0.000244	1	0.000244	1
F8	0.00049	1	0.00049	1	0.00049	1	0.00024	1	0.000244	1	0.000244	1	0.000244	1
F9	0.00049	1	0.00049	1	0.00049	1	0.00049	1	0.000244	1	0.000244	1	0.000244	1
F10	0.00049	1	0.00024	1	0.00024	1	0.00049	1	0.000244	1	0.000244	1	0.000244	1
F11	0.00098	1	0.00049	1	0.00049	1	0.00049	1	0.000244	1	0.000244	1	0.000244	1
F12	0.00489	1	0.00024	1	0.00024	1	0.00024	1	0.000244	1	0.000244	1	0.000244	1
F13	0.00342	1	0.00098	1	0.00098	1	0.00024	1	0.000244	1	0.000244	1	0.000244	1

Figures 8-13 show the convergence graphs of MGO and GOA on classical benchmark functions for dimension=30. Figures 14-17 show the boxplot graphs of MGO and GOA for D={10, 20, 30, 50, 100, 500, and 1000}, respectively. Boxplots show five features of a data set: minimum value, first (25%) quartile, median, third (75%) quartile, and maximum value. Minimum value is the lowest value, excluding outliers (shown at the end of the left). First quarter (25%) shows the twenty-five percent of the scores.

Median is the median marks the midpoint of the data and is represented by the line dividing the box into two parts. Third quarter (75%) is shows the seventy-five percent of the scores. Hence, 25% of the data is above this value. Maximum value shows the highest value excluding outliers. Box plots allow one to quickly identify mean values, distribution of the data set, and signs of variability (https://yalin-dunya.com/2020/06/19/kutu-grafigi-boxplot/).

According to Table 4, MGO achieved superior results than GOA in all classical functions in the best, Mean, and Std comparison criteria. When the time results are examined, it is seen that GOA works in a shorter time than MGO. Worst results are given for informational purposes only and do not mean anything when comparing MGO and GOA. The results obtained in Table 4 are similar in different dimensions (in Tables 5-10). When Table 11 is examined, it is seen that there is a semantic difference between MGO and GOA results in all classical benchmark functions and in all different dimensions. According to Figures 8-13, MGO converged much faster than GOA in all classical benchmark functions.

When Figures 14 - 17 are examined, it is seen that the data distributions are generally not symmetrical. Additionally, the five values (minimum value, first quartile, median, third quartile, maximum value) shown by the box plots are very close to each other. Outliers are observed in some dimension datasets.

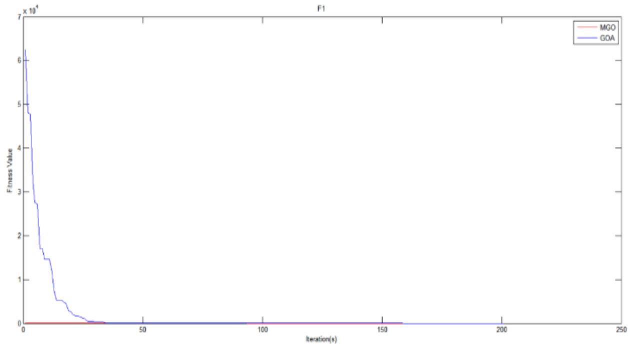


Figure 8. The convergence graphs of MGO and GOA for dimension=30 (F1)

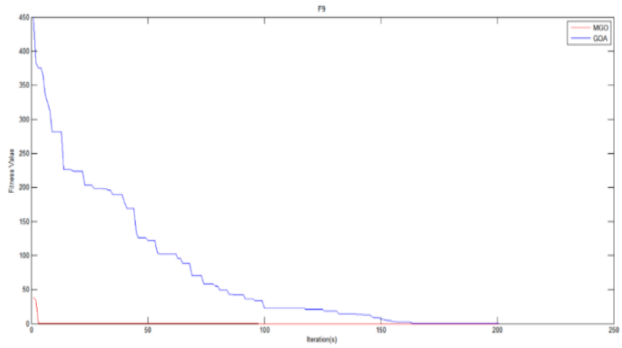


Figure 11. The convergence graphs of MGO and GOA for dimension=30 (F9)

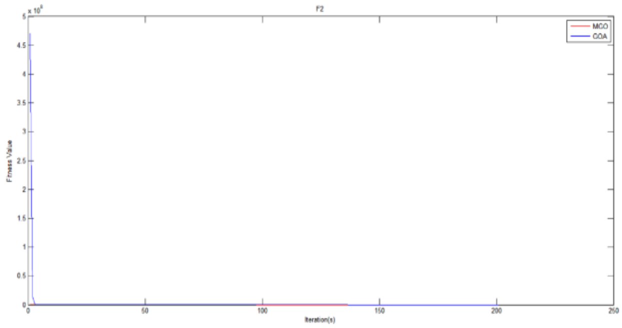


Figure 9. The convergence graphs of MGO and GOA for dimension=30 (F2)

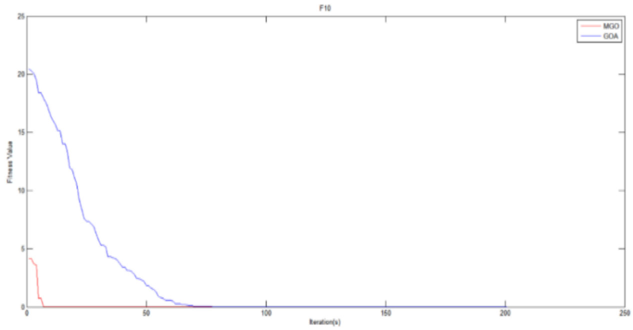


Figure 12. The convergence graphs of MGO and GOA for dimension=30 (F10)

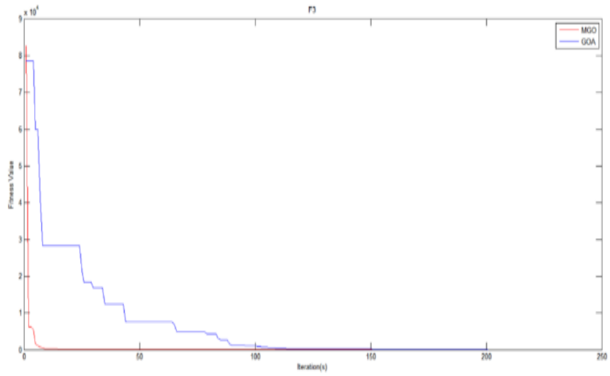


Figure 10. The convergence graphs of MGO and GOA for dimension=30 (F3)

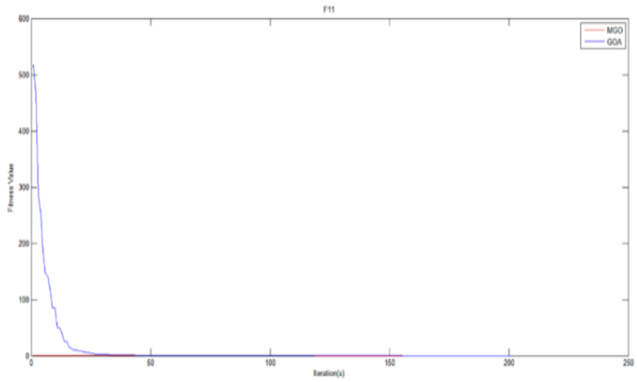
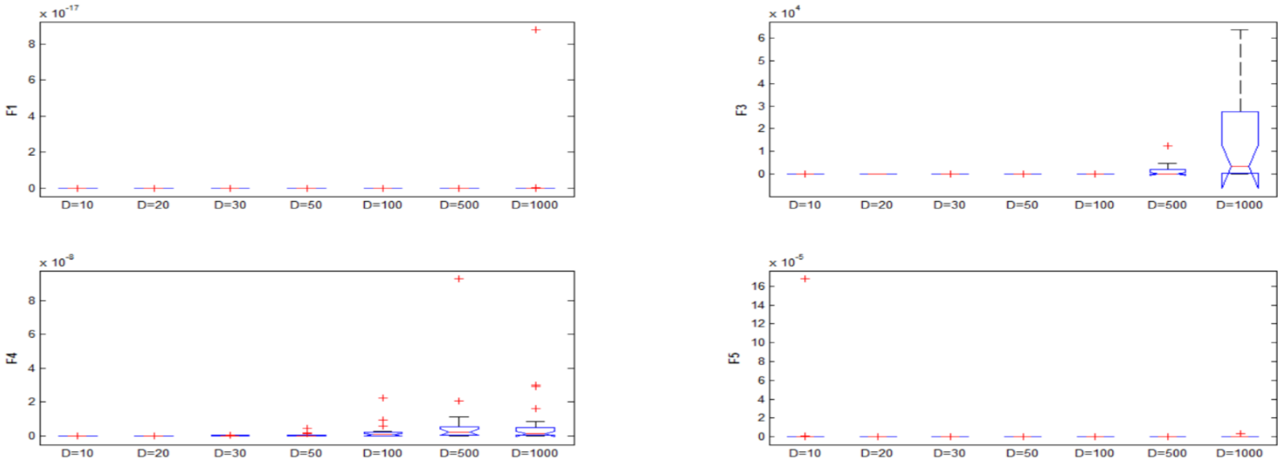


Figure 13. The convergence graphs of MGO and GOA for dimension=30 (F11)



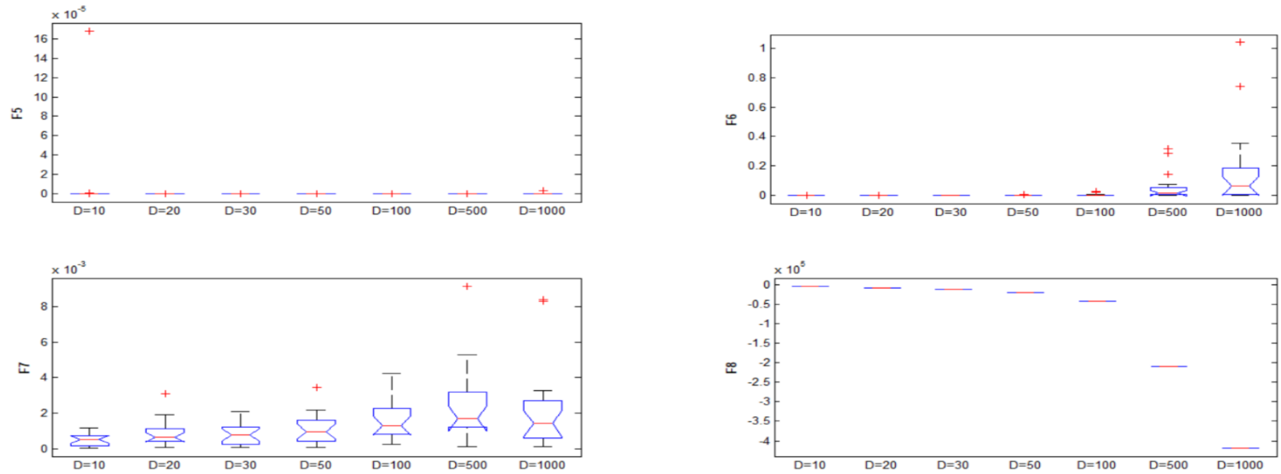


Figure 14. Boxplot graph of MGO for D={10, 20, 30, 50, 100, 500, and 1000}

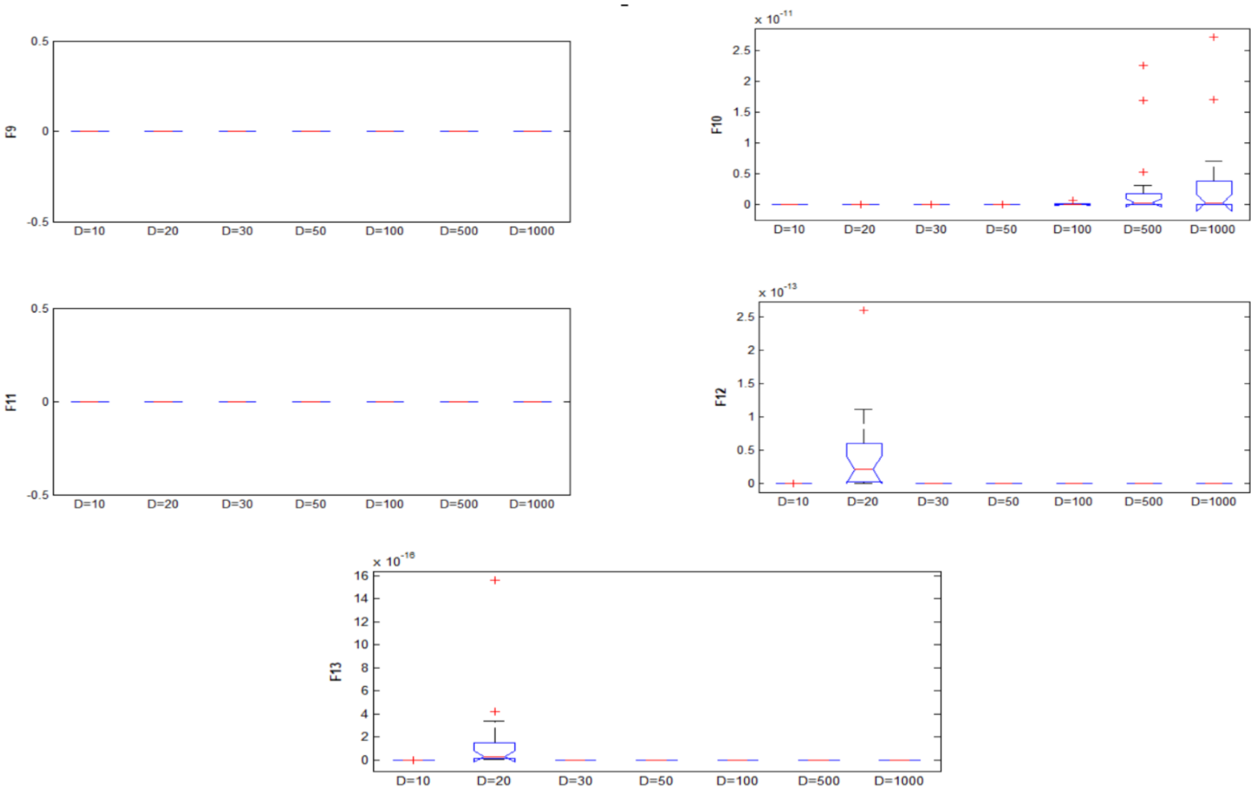
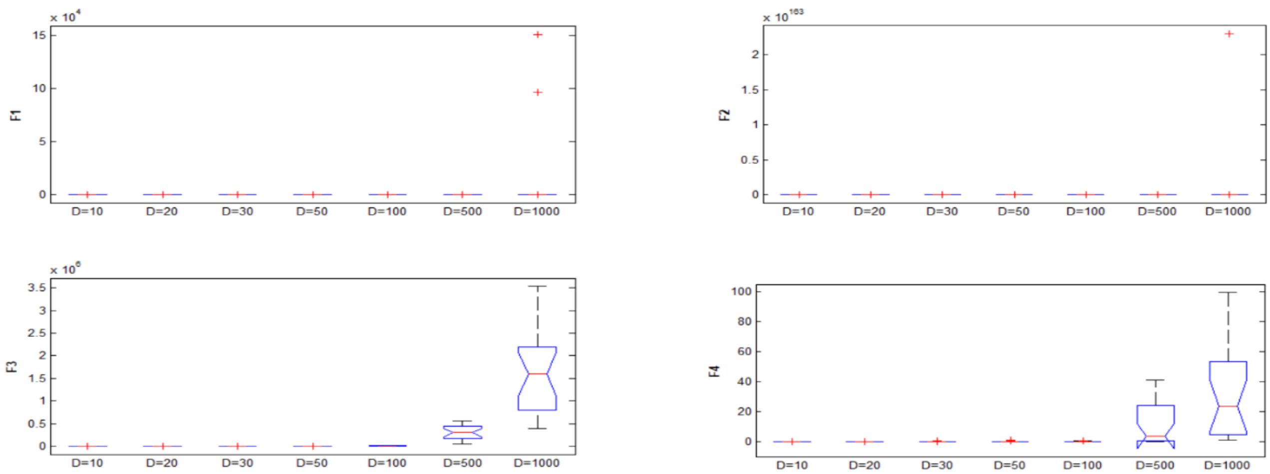


Figure 15. Boxplot graph of MGO for D={10, 20, 30, 50, 100, 500, and 1000}



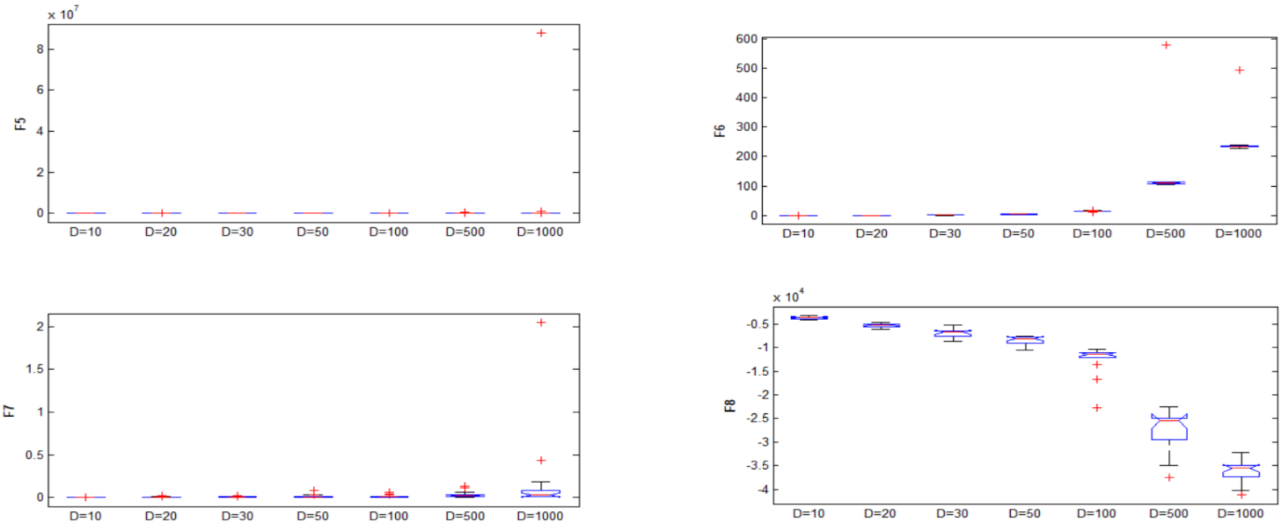


Figure 16. Boxplot graph of GOA for D={10, 20, 30, 50, 100, 500, and 1000}

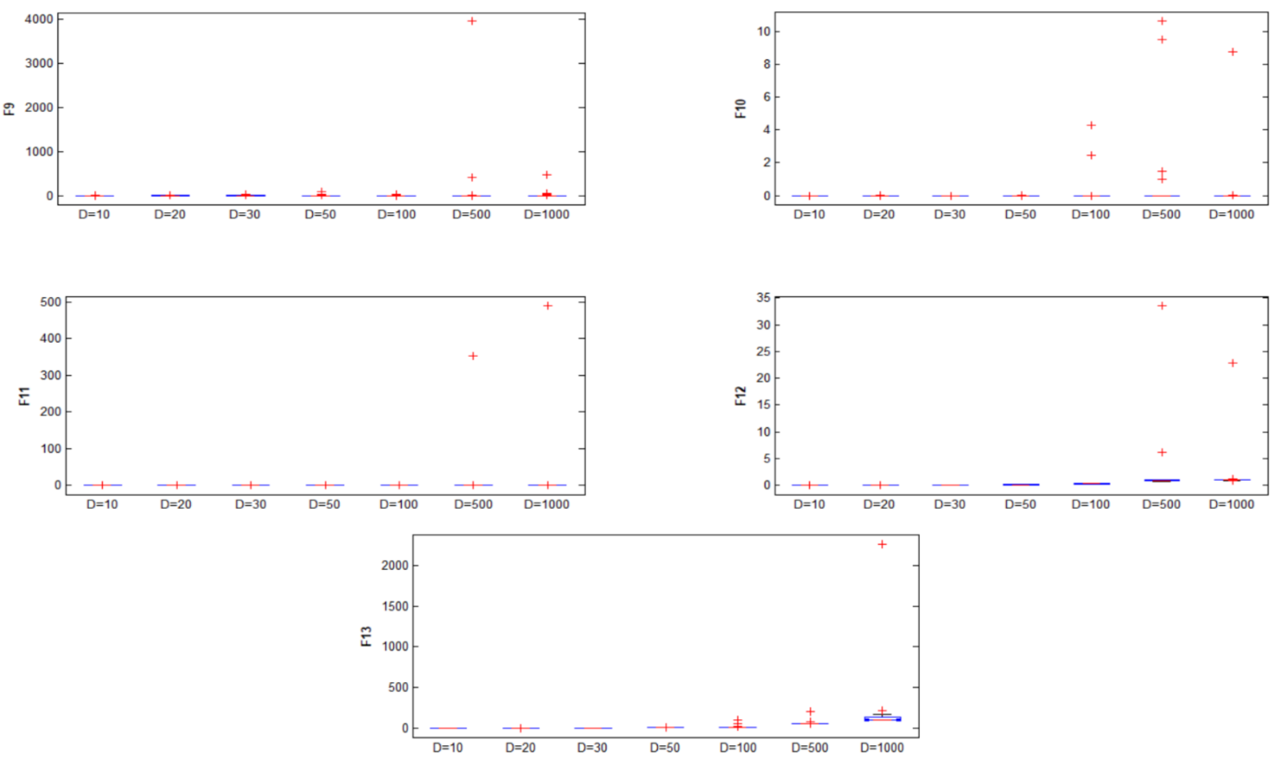


Figure 17. Boxplot graph of GOA for D={10, 20, 30, 50, 100, 500, and 1000}

3.2. The comparison of MGO and GOA on engineering design problems

Algorithms that are successful in classical benchmark functions can often fail to solve real-world problems. That's why MGO and GOA have been shown to be successful in three different engineering problems. The range of all variables in these problems is known and can be controlled [12, 13]. Every problem can be created as a mathematical model [12, 13]. These problems are often difficult to solve due to a lot of calculations and many variables that need to be processed [12, 13]. Three engineering problems are selected in this subsection to measure the success of MGO and GOA algorithms in solving engineering problems.

3.2.1. The comparison of MGO and GOA on pressure vessel design problem

The aim of the pressure vessel design problem is to minimize the total cost of cylindrical pressure vessels [12, 13]. Problem variables: shell thickness (*Ts*), head thickness (*Th*), inner radius (*R*), and container length (*L*) [12, 13]. The mathematical equations of the problem are shown in Equations 24-30 [12, 13]. MGO and GOA were run independently 20 times under equal conditions. The population size was determined as 30 and the maximum number of iterations was determined as 200. Best, worst, average (Mean), standard deviation (Std), and average time (Time) statistical evaluations were made on the results obtained. The statistical results are shown in Table 12. The values of the variable values (T_s, T_h, R, and L) in the case of the best cost result are shown in Table 13. Figure 18 shows the schematic view of the the pressure vessel design problem [12, 13]. Figure 19 shows the convergence graphs of MGO and GOA for the pressure vessel design problem.

According to Table 12, while MGO achieved the best results, GOA also achieved the best results in terms of mean and standard deviation. Additionally, GOA worked in a shorter time. According to Table 13, in 20 independent studies, MGO achieved a better result than GOA and obtained the corresponding problem variables. Thus, MGO ranked first in the rank order. According to Figure 19, although MGO achieved a worse fitness at first, it converged to the best result with a faster convergence than GOA. GOA, on the other hand, achieved slower convergence in each iteration.

Variables:

$$\vec{x} = [x_1\ x_2\ x_3\ x_4] = [T_s, T_h\ R\ L] \tag{24}$$

$$\text{Minimize } f(\vec{x}) = 0.6224x_1x_2x_3 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \tag{25}$$

Subject to:

$$g_1(\vec{x}) = -x_1 + 0.0193x_3 \le 0, \tag{26}$$

$$g_2(\vec{x}) = -x_2 + 0.00954x_3 \le 0, \tag{27}$$

$$g_3(\vec{x}) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0 \tag{28}$$

$$g_4(\vec{x}) = x_4 - 240 \le 0. \tag{29}$$

Variable range:

$$0 \le x_1, x_2 \le 99,\ \ 10 \le x_3, x_4 \le 200 \tag{30}$$

Table 12. The statistical results of MGO and GOA for the pressure vessel design problem

Algorithm	Best	Worst	Mean	Std	Time
MGO	5888.0874	7318.982	6422.1588	506.88686	2.765137
GOA	5901.282591	6226.880046	5966.696622	73.00917999	0.699607

Table 13. The comparison results of MGO and GOA for the pressure vessel design problem

Algorithm	T_s	T_h	R	L	Best cost	Rank
MGO	0.77977687	0.38544411	40.4029465	198.8433	5888.0874	1
GOA	0.778731	0.389645	40.34135	199.7133	5901.282591	2

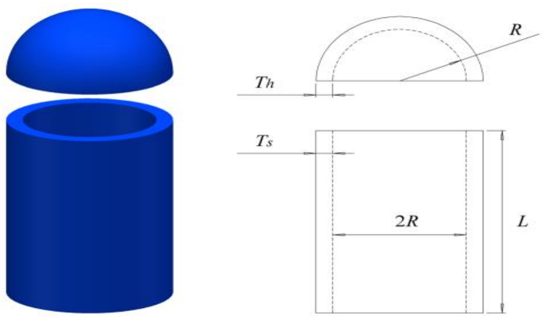


Figure 18. The The schematic view of the pressure vessel design problem [12, 13]

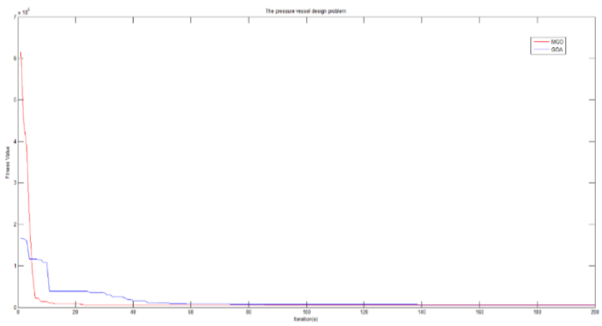


Figure 19. The convergence graphs of MGO and GOA for the pressure vessel design problem

3.2.2. The comparison of MGO and GOA on welded beam design problem

The goal of this problem is to obtain the minimum weight under four constraint conditions [12, 13]. Problem variables: weld width h, connecting beam length l, beam height t, and connecting beam thickness b [12, 13]. The mathematical equations of the problem are shown in Equations 31-46 [12, 13]. MGO and GOA were run independently 20 times under equal conditions. The population size was determined as 30 and the maximum number of iterations was determined as 200.

Best, worst, average (Mean), standard deviation (Std), and average time (Time) statistical evaluations were made on the results obtained. The statistical results are shown in Table 14. The values of the variable values (*h*, *l*, *t*, and *b*) in the case of the best cost result are shown in Table 15. Figure 20 shows the schematic view of the the welded beam design problem [12, 13]. Figure 21 shows the convergence graphs of MGO and GOA for the welded beam design problem.

According to Table 14, while GOA achieved the best results, MGO also achieved the best results in terms of mean and standard deviation. GOA worked in a shorter time than MGO. According to Table 15, in 20 independent studies, GOA achieved a better result than MGO and obtained the corresponding problem variables. Thus, GOA ranked first in the rank order. According to Figure 21, GOA converged to the best result with a faster convergence than MGO.

Variables:

$$\vec{x} = [x_1\ x_2\ x_3\ x_4] = [h\ l\ t\ b] \tag{31}$$

$$Minimize\ f(\vec{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4 + (14.0 + x_2) \tag{32}$$

Subject to:

$$g_1(\vec{x}) = \tau(\vec{x}) - \tau_{max} \le 0, \tag{33}$$

$$g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{max} \le 0, \tag{34}$$

$$g_3(\vec{x}) = \delta(\vec{x}) - \delta_{max} \le 0, \tag{35}$$

$$g_4(\vec{x}) = x_1 - x_4 \le 0, \tag{36}$$

$$g_5(\vec{x}) = P - P_c(\vec{x}) \le 0, \tag{37}$$

$$g_6(\vec{x}) = 0.125 - x_1 \le 0, \tag{38}$$

$$g_7(\vec{x}) = 1.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 0.5 \le 0, \tag{39}$$

where:

$$\tau(\vec{x}) = \sqrt{(\tau')^2\ 2\tau'\tau''\frac{x_2}{2R} + (\tau'')},\ \tau' = \frac{P}{\sqrt{2x_1x_2}},\tau'' = \frac{MR}{J}, \tag{40}$$

$$M = P\left(L + \frac{x_2}{2}\right),\quad R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2},\quad \sigma(\vec{x}) = \frac{6PL}{x_4x_3^2}, \tag{41}$$

$$J = 2\left\{\sqrt{2x_1x_2}\left[\frac{x_2^2}{4}\left(\frac{x_1+x_3}{2}\right)^2\right]\right\}, \delta(\vec{x}) = \frac{6PL^3}{Ex_4x_3^2}, \tag{42}$$

$$P_c(\vec{x}) = \frac{4.01E\sqrt{\frac{x_3^2x_4^6}{0}}}{L^2}, \left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right), \left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right), \tag{43}$$

$$P = 6000lb, L = 14\ in, \delta_{max} = 0.25\ in, E = 30 \times 10^6\ psi, \tag{44}$$

$$\tau_{max} = 13600\ psi,\ and\ \sigma_{max} = 30000\,psi \tag{45}$$

Variable range:

$$0.1 \le x_1 \le 2,\ 0.1 \le x_4 \le 2,\ 0.1 \le x_1 \le 10,\ 0.1 \le x_1 \le 10 \tag{46}$$

Table 14. The statistical results of MGO and GOA for the welded beam design problem

Algorithm	Best	Worst	Mean	Std	Time
MGO	2.397591	2.903343	2.507813	0.141959	1.97952
GOA	2.386234	3.334676	2.569828	0.255182	0.754791

Table 15. The comparison results of MGO and GOA for the welded beam design problem

Algorithm	*h*	*l*	*t*	*b*	Best cost	Rank
MGO	0.243847	6.026648	8.517861	0.243909	2.397591	2
GOA	0.243791	6.216256	8.324413	0.244319	2.386234	1

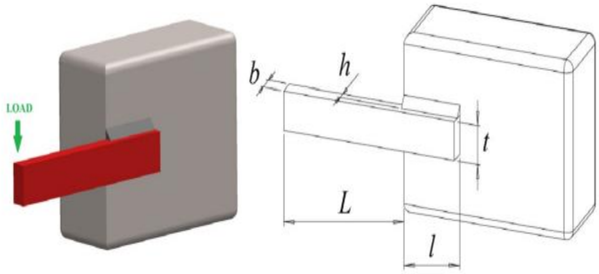


Figure 20. The schematic view of the the welded beam design problem [12, 13]

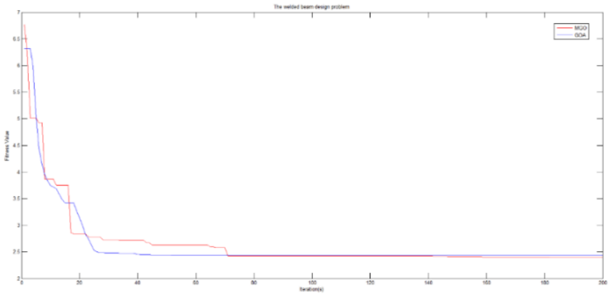


Figure 21. The convergence graphs of MGO and GOA for the welded beam design problem

3.2.3. The comparison of MGO and GOA on tension/compression spring design problem

The aim of the problem is to obtain the minimum spring mass with the existing variables and constraints [12, 13]. Problem variables: coil diameter *d*, average coil diameter *D*, and efective coil number *N* [12, 13]. The mathematical equations of the problem are shown in Equations 47-53 [12, 13]. MGO and GOA were run independently 20 times under equal conditions. The population size was determined as 30 and the maximum number of iterations was determined as 200. Best, worst, average (Mean), standard deviation (Std), and average time (Time) statistical evaluations were made on the results obtained.

The statistical results are shown in Table 16. The values of the variable values (*d*, *D*, and *N*) in the case of the best cost result are shown in Table 17. Figure 22 shows schematic view the tension/compression spring design problem [12, 13]. Figure 23 shows the convergence graphs of MGO and GOA for the tension/compression spring design problem.

According to Table 16, MGO achieved the best results in terms of best, mean, and standard deviation. GOA worked in a shorter time than MGO. According to Table 17, in 20 independent studies, MGO achieved a better result than GOA and obtained the corresponding problem variables. Thus, MGO ranked first in the rank order. According to Figure 23, MGO converged to the best result with a faster convergence than GOA.

Variables:

$$\vec{x} = [x_1\ x_2\ x_3] = [d\ D\ N] \quad (47)$$

$$Minimize\ f(\vec{x}) = (x_3 + 2) \times x_2 \times x_1^2 \quad (48)$$

Subject to:

$$g_1(\vec{x}) = 1 - \frac{x_3 \times x_2^3}{71785 \times x_1^4} \leq 0, \quad (49)$$

$$g_2(\vec{x}) = \frac{4 \times x_2^2 - x_1 \times x_2}{12566 \times x_1^4} + \frac{1}{5108 \times x_1^2} - 1 \leq 0, \quad (50)$$

$$g_3(\vec{x}) = 1 - \frac{140.45 \times x_1}{x_2^2 \times x_3} \leq 0, \quad (51)$$

$$g_4(\vec{x}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0, \quad (52)$$

Variable range:

$$0.05 \leq x_1 \leq 2.0,\ 0.25 \leq x_2 \leq 1.3,\ 2.0 \leq x_3 \leq 15.0 \quad (53)$$

Table 16. The statistical results of MGO and GOA for the tension/compression spring design problem

Algorithm	Best	Worst	Mean	Std	Time
MGO	0.012682	0.017774	0.013348	0.001234	1.290039
GOA	0.01269	0.017423	0.013678	0.001482	0.617529

Table 17. The comparison results of MGO and GOA for the tension/compression spring design problem

Algorithm	*d*	*D*	*N*	Best cost	Rank
MGO	0.051513	0.352257	11.56701	0.012682	1
GOA	0.052776	0.383432	9.882135	0.01269	2

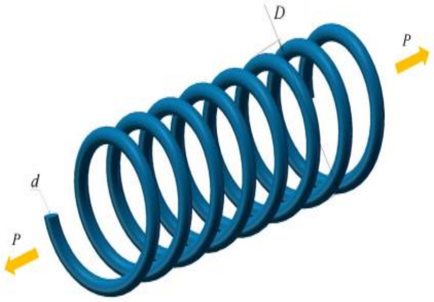


Figure 22. The schematic view of the tension/compression spring design problem [12, 13]

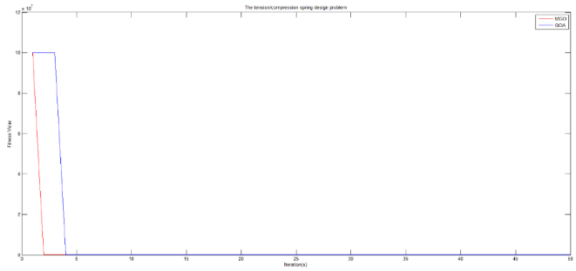


Figure 23. The convergence graphs of MGO and GOA for the tension/compression spring design problem

3.3. The comparisons of the MGO, GOA, and other algorithms on classical test Functions

In this subsection, three different heuristic algorithms newly proposed in recent years, MGO and GOA, are compared. Thus, the success of MGO and GOA was compared with the literature. Algorithms selected from the literature are as follows: Golden Search Optimization algorithm (GSO) [14], Crayfsh Optimization Algorithm (COA) [13], and Zebra Optimization Algorithm (ZOA) [12]. The codes of the MGO, GOA, GSO, COA, and ZOA algorithms were obtained from the mathworks library (https://ww2.mathworks.cn/en/).

All algorithms were compared under equal conditions on 13 classical benchmarks. The population size was determined as 30, the maximum iteration was determined as 200, and the dimension was determined as 30. Each algorithm was run independently 20 times. Best, mean, standard deviation (Std), and average time (Time) calculations were made on the results. The results are shown in Tables 18-19. Convergence graphs of the algorithms are shown in Figures 24-31.

When the average results are examined according to Table 18, the most successful algorithms are MGO and COA. Both algorithms showed superior success in 7 out of 13 functions. MGO and COA are followed by ZOA and GSO, respectively. The most unsuccessful algorithm was GOA. When Table 19 is examined, a similar situation can be seen. While MGO was especially successful in the F5, F6, F8, F9, F10, F11, F12, and F13 functions, COA was especially successful in the F1, F2, F3, F4, F9, F10, and F11 functions. Total average time results are listed in Table 19.

In this case, the fastest running algorithm was ZOA. It was followed by GSO and COA. The longest running algorithm was MGO. When the convergence graphs are examined, it is observed that GOA and GSO converge slowly, while MGO and COA converge faster.

Table 18. The comparisons results of the MGO, GOA, and other algorithms (D=30)

F_ID	*MGO*		*GOA*		*GSO*		*COA*		*ZOA*	
	Mean	***Std***	***Mean***	***Std***	***Mean***	***Std***	***Mean***	***Std***	***Mean***	***Std***
F1	8.630e-29	3.31e-28	2.40E-05	7.4E-05	3.00E-111	9.86E-111	0.00E+00	0.00E+00	4.10E-95	1.53E-94
F2	2.38e-17	4.04e-17	2.17E-08	9.36E-08	4.40E-57	1.33E-56	8.51E-156	0.00E+00	1.50E-51	4.10E-51
F3	2.87E-03	6.64E-03	4.93E+01	1.55E+02	8.21E-108	3.56E-107	0.00E+00	0.00E+00	1.06E-58	3.12E-58
F4	9.14e-11	1.05e-10	5.02E-02	1.38E-01	5.23E-46	2.27E-45	1.34E-160	0.00E+00	5.29E-44	1.33E-43
F5	7.48e-11	1.94e-10	2.73E+01	6.35E-01	2.42E+01	1.92E+01	2.84E+01	4.43E-01	2.87E+01	1.65E-01
F6	2.03e-05	2.43e-05	7.91E-01	2.92E-01	2.07E+00	1.94E+00	1.94E+00	6.55E-01	3.63E+00	5.29E-01
F7	8.18E-04	5.89E-04	7.68E-03	3.96E-03	4.11E-04	4.82E-04	2.41E-04	2.46E-04	2.19E-04	1.57E-04
F8	-1.26E+04	2.32E-04	-6.84E+03	7.55E+02	-1.03E+04	5.58E+02	-6.96E+03	9.87E+02	-6.26E+03	5.98E+02
F9	0.00E+00	0.00E+00	2.66E+00	7.33E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F10	6.22e-15	9.50e-15	2.56E-05	1.11E-04	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00
F11	0.00E+00	0.00E+00	2.08E-03	6.50E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F12	6.00e-13	1.09e-12	3.51E-02	1.30E-02	5.93E-02	8.34E-02	1.24E-01	1.16E-01	2.44E-01	9.99E-02
F13	7.93e-15	1.20e-14	5.48E-01	1.68E-01	2.52E-01	2.82E-01	2.56E+00	2.32E-01	2.26E+00	3.08E-01

Table 19. The comparisons results of the MGO, GOA, and other algorithms (D=30)

F_ID	*MGO*		*GOA*		*GSO*		*COA*		*ZOA*	
	Best	***Time***	***Best***	***Time***	***Best***	***Time***	***Best***	***Time***	***Best***	***Time***
F1	3.99e-37	1.089	1.02E-22	0.508	5.43E-170	0.094	0.00E+00	0.454	3.72E-102	0.053
F2	5.43e-19	1.424	5.49E-14	0.713	2.91E-86	0.096	2.04E-236	0.512	9.39E-55	0.061
F3	5.27e-07	2.816	3.85E-02	1.336	1.59E-158	0.311	0.00E+00	1.189	5.22E-68	0.568
F4	3.25e-13	1.290	8.45E-07	0.750	3.93E-69	0.122	1.26E-211	0.767	2.14E-46	0.115
F5	3.60e-16	1.530	2.65E+01	0.907	1.37E+00	0.323	2.73E+01	0.711	2.84E+01	0.133
F6	6.10e-08	1.372	3.29E-01	0.822	4.69E-04	0.422	5.00E-01	0.497	2.30E+00	0.104
F7	7.77e-05	1.491	3.75E-03	0.867	6.19E-06	0.471	1.17E-05	0.518	1.96E-05	0.182
F8	-1.26E+04	1.431	-8.64E+03	0.814	-1.10E+04	0.207	-8.36E+03	0.574	-7.39E+03	0.107
F9	0.00E+00	1.542	0.00E+00	0.827	0.00E+00	0.198	0.00E+00	0.748	0.00E+00	0.125
F10	8.88e-16	1.559	6.07E-12	0.798	8.88E-16	0.186	8.88E-16	0.860	8.88E-16	0.160
F11	0.00E+00	1.514	0.00E+00	0.842	0.00E+00	0.193	0.00E+00	0.576	0.00E+00	0.159
F12	3.52e-15	2.017	1.24E-02	1.082	2.99E-04	0.300	3.30E-02	0.777	7.26E-02	0.400
F13	7.53e-17	2.045	2.70E-01	1.054	7.44E-03	0.306	2.06E+00	0.860	1.62E+00	0.411
Total Time:		2.11E+01		1.13E+01		3.23E+00		9.04E+00		2.58E+00
Rank:		5		4		2		3		1

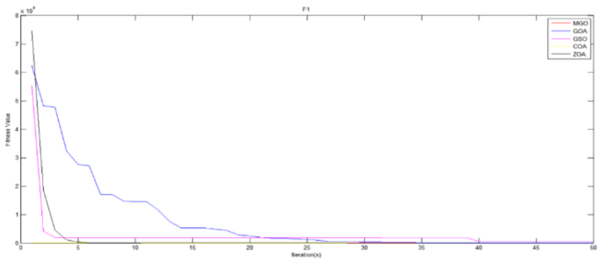


Figure 24. The convergence graphs of MGO, GOA, GSO, COA, and ZOA for dimension=30 (F1)

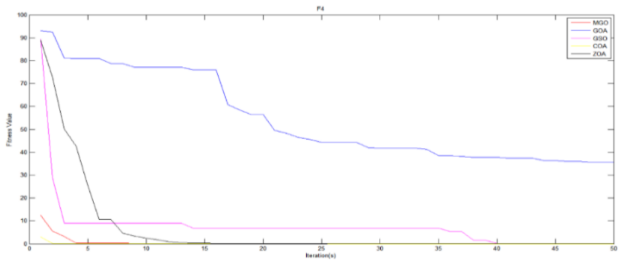


Figure 27. The convergence graphs of MGO, GOA, GSO, COA, and ZOA for dimension=30 (F4)

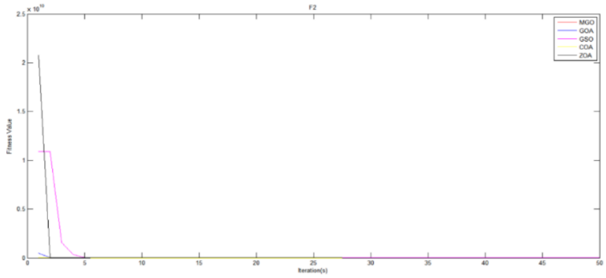


Figure 25. The convergence graphs of MGO, GOA, GSO, COA, and ZOA for dimension=30 (F2)

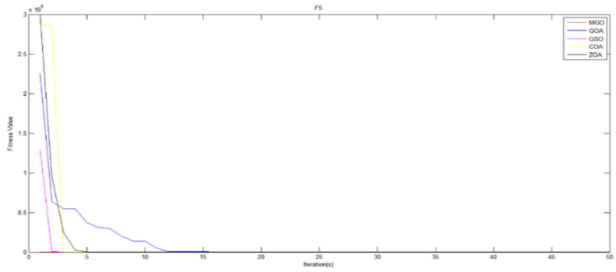


Figure 28. The convergence graphs of MGO, GOA, GSO, COA, and ZOA for dimension=30 (F5)

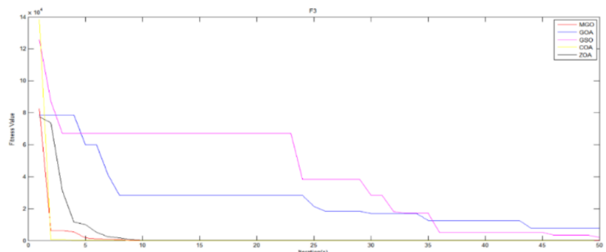


Figure 26. The convergence graphs of MGO, GOA, GSO, COA, and ZOA for dimension=30 (F3)

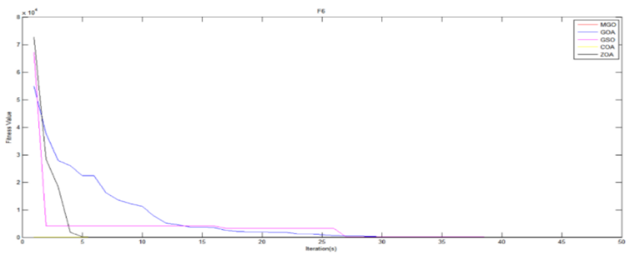


Figure 29. The convergence graphs of MGO, GOA, GSO, COA, and ZOA for dimension=30 (F6)

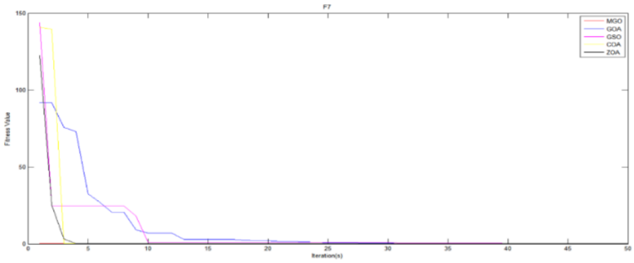


Figure 30. The convergence graphs of MGO, GOA, GSO, COA, and ZOA for dimension=30 (F7)

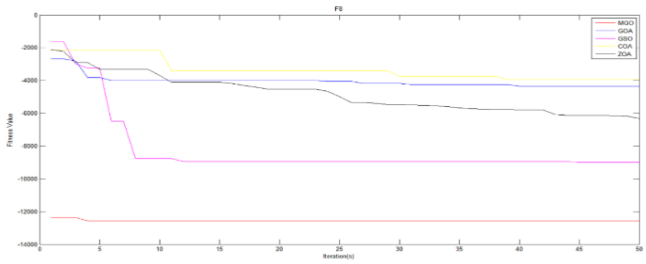


Figure 31. The convergence graphs of MGO, GOA, GSO, COA, and ZOA for dimension=30 (F8)

3.4. The parameter analysis for GOA and MGO algorithm

In this subsection, a population analysis was performed for MGO and GOA algorithms. In addition, *PSRs* and *S* parameters, which are fixed parameters of GOA, were analyzed. There are no special fixed parameters for MGO. The discovery and exploitation abilities of these parameters on the algorithms are discussed.

To analyze the effect of population size on MGO and GOA, ten different values were examined (N={10, 20, 30, 40, 50, 60, 70, 80, 90, and 100}) on the classic benchmarks. MGO and GOA were run independently 20 times, with a dimension of 30 and a maximum iteration of 200. Average calculations were made on the results obtained. The results are shown in Tables 20-21. According to the results, as the population size increased, the performance of MGO and GOA increased. Inversely proportional to this, working time has also increased.

In this subsection, two fixed parameter analyzes that affect GOA's exploration and exploitation capabilities have been carried out. The first of these is the *PSRs* parameter setting. The effect of nine different values on GOA (*PSRs*= {0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, and 0.50}) for the *PSRs* parameter was examined. GOA was run independently 20 times, with a dimension of 1000, a maximum iteration of 200, and a population size of 30. The mean of the results obtained are calculated and the results are shown in Table 22. According to Table 22, the most successful results were obtained when the *PSRs* value was 0.50. It was later followed by 0.35. The *PSRs* value was taken as 0.34 by Agushaka et al. in their original paper [5]. The findings obtained are parallel to the original paper. In this study, the *PSRs* parameter value was taken as 0.34 in the experimental part of the study.

The second parameter analysis performed on GOA is the *S* parameter value. the effects of nine different values for the *S* parameter on GOA were examined. The mean of the results obtained are calculated and the results are shown in Table 23. According to Table 23, the most successful results were obtained when the S value was 0.10. It was later followed by 0.90. The *S* value was taken as 0.88 by Agushaka et al. in their original paper. It has been observed that there is no full agreement between the value suggested in the original paper and the value found for the *S* value. In this study, in order to remain faithful to the original structure of GOA, the *S* parameter value was taken as 0.88 in the experimental part of the study.

4. Conclusion

In this study, two newly proposed heuristic algorithms in recent years were examined. These algorithms are mountain gazelle optimization (MGO) and gazelle optimization algorithm (GOA). both algorithms were inspired by the social lifestyle of gazelles. Due to this similarity, they are often confused in the literature and seen as the same algorithms. This study was carried out to eliminate this confusion. Four main factors in the life of mountain gazelles are used in the MGO mathematical model.
These are single male herds, natal herds, solitary, territorial males and migrate in search of food. MGO realizes its exploration and exploitation abilities with these four groups of mountain gazelles. The GOA model was inspired by the behavior of gazelles to escape from predators, reach safe environments and graze in safe environments. While the grazing behavior of gazelles in safe environments was used for the

exploitation ability of the GOA, the behavior of escaping from predators was used for the exploration ability of the GOA. MGO and GOA were run on 13 classical unimodal and multimodal benchmark functions in seven different dimensions (10, 20, 30, 50, 100, 500, 1000) and their success was compared. According to the results, MGO is more successful than GOA in all dimensions. GOA, on the other hand, works faster than MGO. Then, MGO and GOA were tested on 3 different engineering design problems.

While MGO was more successful in the tension/compression spring design problem and welded beam design problems, GOA achieved better results in the pressure vessel design problem. The success of MGO and GOA has been compared with 3 different algorithms (GSO, COA, and ZOA) that have been proposed in the literature in recent years. While MGO competes with literature algorithms, GOA lags behind the literature.

In future studies, it is planned to hybridize MGO and GOA (modified MGO-GOA) to obtain a more successful heuristic algorithm. It is planned to combine the superior aspects of both heuristic algorithms and eliminate the negative aspects.

Table 20. The population size analysis for MGO on classical benchmark functions (D=30, $Iter_{max}$= 200, *PSRs* = 0.34, and *S*=0.88) (Mean)

F_ID	*MGO*									
	N=10	***N=20***	***N=30***	***N=40***	***N=50***	***N=60***	***N=70***	***N=80***	***N=90***	***N=100***
	Mean	***Mean***	***Mean***	***Mean***	***Mean***	***Mean***	***Mean***	***Mean***	***Mean***	***Mean***
F1	3.68E-20	5.76E-26	8.630e-29	2.46E-32	3.09E-34	1.96E-34	1.15E-36	1.14E-38	3.07E-39	7.36E-41
F2	4.15E-14	2.36E-15	2.38e-17	6.38E-19	1.17E-19	3.70E-21	1.74E-21	4.47E-22	3.82E-23	1.11E-23
F3	6.90E-01	2.36E-02	2.87E-03	1.57E-04	3.30E-04	1.71E-03	3.28E-05	3.15E-05	6.76E-06	1.75E-06
F4	4.45E-07	3.84E-10	9.14e-11	1.57E-11	3.95E-12	3.76E-13	1.19E-13	1.15E-13	1.12E-14	1.65E-14
F5	2.46E-06	2.02E-08	7.48e-11	4.91E-11	6.80E-12	1.38E-11	3.50E-11	1.95E-13	1.92E-12	8.12E-14
F6	3.16E-03	2.36E-04	2.03e-05	5.45E-06	3.18E-07	3.72E-08	2.19E-07	1.22E-08	5.13E-09	1.31E-09
F7	2.17E-03	1.37E-03	8.18E-04	7.43E-04	5.26E-04	6.18E-04	6.02E-04	3.76E-04	3.31E-04	3.06E-04
F8	-1.26E+04	-1.26E+04	-1.26E+04	-1.26E+04	-1.26E+04	-1.26E+04	-1.26E+04	-1.26E+04	-1.26E+04	-1.26E+04
F9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F10	2.13E-11	4.85E-14	6.22e-15	2.84E-15	2.31E-15	2.31E-15	1.60E-15	1.60E-15	1.60E-15	8.88E-16
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F12	5.75E-09	9.54E-12	6.00e-13	8.76E-14	6.01E-15	1.34E-15	7.88E-16	2.32E-16	8.11E-17	4.37E-17
F13	1.26E-10	2.17E-13	7.93e-15	2.75E-16	1.35E-16	3.73E-18	2.00E-18	2.61E-19	1.21E-19	1.60E-19
Total:	-12599.30	-12599.97	-12600.00	-12600.00	-12600.00	-12600.00	-12600.00	-12600.00	-12600.00	-12600.00
Rank:	10	9	7	8	5	6	4	3	2	1

Table 21. The population size analysis for GOA on classical benchmark functions (D=30, $Iter_{max}$ **=** 200, *PSRs* = 0.34, and *S*=0.88) (Mean)

F_ID	*GOA*									
	N=10	***N=20***	***N=30***	***N=40***	***N=50***	***N=60***	***N=70***	***N=80***	***N=90***	***N=100***
	Mean	***Mean***	***Mean***	***Mean***	***Mean***	***Mean***	***Mean***	***Mean***	***Mean***	***Mean***
F1	1.19E-03	8.25E-07	2.40E-05	6.64E-06	1.36E-06	1.24E-14	9.35E-10	2.80E-08	6.63E-08	4.95E-08
F2	1.97E-04	1.55E-05	2.17E-08	1.13E-08	5.37E-06	3.57E-06	1.03E-06	2.91E-06	3.24E-06	2.02E-06
F3	4.07E+01	5.13E+01	4.93E+01	5.87E+01	8.86E+01	6.49E+01	5.86E+01	5.40E+01	5.08E+01	1.35E+01
F4	3.77E-01	4.37E-01	5.02E-02	2.80E-02	3.83E-02	1.02E-01	2.05E-04	1.36E-04	9.45E-04	7.32E-02
F5	2.92E+01	2.73E+01	2.73E+01	2.68E+01	2.66E+01	2.68E+01	2.62E+01	2.64E+01	2.61E+01	3.01E+01
F6	2.03E+00	8.48E-01	7.91E-01	4.58E-01	4.82E-01	4.03E-01	4.83E-01	3.64E-01	3.38E-01	2.72E-01
F7	2.25E-02	7.53E-03	7.68E-03	4.88E-03	5.81E-03	5.62E-03	3.82E-03	5.26E-03	2.91E-03	3.93E-03
F8	-6.04E+03	-6.42E+03	-6.84E+03	-7.04E+03	-6.85E+03	-6.96E+03	-7.12E+03	-7.30E+03	-7.57E+03	-7.17E+03
F9	1.09E+01	4.92E+00	2.66E+00	4.09E+00	4.32E+00	2.62E+00	4.57E+00	1.57E+00	2.94E+00	6.09E+00
F10	2.99E-05	2.01E-06	2.56E-05	5.33E-05	1.36E-04	8.64E-06	8.27E-07	2.11E-04	9.00E-06	7.29E-06
F11	1.01E-02	1.25E-03	2.08E-03	5.52E-04	7.29E-05	9.40E-04	1.19E-03	7.17E-04	5.02E-04	3.25E-03
F12	8.57E-02	4.01E-02	3.51E-02	2.81E-02	2.96E-02	2.50E-02	2.12E-02	2.20E-02	1.99E-02	1.40E-02
F13	1.36E+00	8.68E-01	5.48E-01	4.49E-01	4.08E-01	3.78E-01	3.24E-01	2.58E-01	3.27E-01	2.74E-01
Total:	-5.96E+03	-6.33E+03	-6.76E+03	-6.95E+03	-6.73E+03	-6.86E+03	-7.03E+03	-7.22E+03	-7.49E+03	-7.12E+03
Rank:	5	10	8	6	9	7	4	2	1	3

Table 22. The *PSRs* parameter analysis for GOA algorithm on classical benchmark functions (N=30, D=1000, $Iter_{max}$ = 200, and S = 0.88)

F_ID	GOA								
	PSRs=0.10	***PSRs=0.15***	***PSRs=0.20***	***PSRs=0.25***	***PSRs=0.30***	***PSRs=0.35***	***PSRs=0.40***	***PSRs=0.45***	***PSRs=0.50***
	Mean	***Mean***	***Mean***	***Mean***	***Mean***	***Mean***	***Mean***	***Mean***	***Mean***
F1	2.66E+03	3.70E+02	1.12E-01	5.98E-01	1.38E+01	1.09E+02	3.48E+01	2.55E+01	4.58E+01
F2	2.07E+88	2.94E+166	1.57E+122	1.26E+110	6.10E+225	8.54E+02	4.49E+142	4.50E+202	5.73E+02
F3	9.46E+05	9.47E+05	1.03E+06	1.18E+06	8.81E+05	1.47E+06	1.50E+06	1.64E+06	2.42E+06
F4	9.47E+00	1.98E+01	2.14E+01	2.00E+01	2.59E+01	2.05E+01	2.83E+01	2.71E+01	3.92E+01
F5	6.95E+05	2.20E+03	8.46E+05	3.29E+05	7.45E+03	5.89E+05	3.83E+04	1.08E+03	6.10E+04
F6	6.55E+03	2.34E+02	3.15E+03	6.34E+03	2.72E+02	2.88E+02	2.36E+02	2.37E+02	1.80E+03
F7	2.68E+01	2.64E-01	1.68E+01	7.21E-02	6.57E-02	1.18E-01	1.41E+01	9.14E+00	8.06E+01
F8	-3.47E+04	-3.67E+04	-3.86E+04	-3.98E+04	-4.06E+04	-3.62E+04	-4.07E+04	-3.69E+04	-3.92E+04
F9	1.70E+00	5.16E+01	1.60E+02	1.94E+01	1.86E+01	1.58E+02	1.58E+01	1.67E+02	3.19E+02
F10	2.66E-01	7.02E-01	1.71E-01	2.70E-03	2.12E-01	7.21E-01	3.67E-01	3.50E-02	1.50E-01
F11	2.05E+00	9.27E-01	3.90E+01	2.78E+01	2.19E+01	2.40E+01	5.37E+01	6.14E+01	2.17E+00
F12	1.50E+00	1.01E+00	5.31E+04	2.94E+03	1.33E+00	1.49E+00	2.99E+04	2.15E+00	7.79E+01
F13	1.19E+02	1.13E+05	1.07E+02	1.15E+02	1.07E+02	8.50E+05	3.09E+06	4.98E+05	1.35E+02
Total:	2.07E+88	2.94E+166	1.57E+122	1.26E+110	6.10E+225	2.87E+06	4.49E+142	4.50E+202	2.44E+06
Rank:	3	7	5	4	9	2	6	8	1

Table 23. The *S* parameter analysis for GOA algorithm on classical benchmark functions (N=30, D=1000, $Iter_{max}$ = 200, and *PSRs* = 0.34)

F_ID	GOA								
	S=0.10	***S=0.20***	***S=0.30***	***S=0.40***	***S=0.50***	***S=0.60***	***S=0.70***	***S=0.80***	***S=0.90***
	Mean	***Mean***	***Mean***	***Mean***	***Mean***	***Mean***	***Mean***	***Mean***	***Mean***
F1	3.32E+03	1.98E+02	4.11E+03	3.25E+03	3.04E-04	9.38E+03	1.05E+03	1.62E+02	4.25E-02
F2	8.90E+02	1.09E+29	1.14E+79	6.50E+142	3.41E+131	1.79E+161	1.58E+10	3.99E+67	8.15E+02
F3	1.14E+06	9.67E+05	9.64E+05	1.16E+06	1.16E+06	1.24E+06	1.47E+06	1.35E+06	1.29E+06
F4	6.40E+00	2.25E+01	1.59E+01	1.92E+01	1.53E+01	2.62E+01	1.90E+01	3.30E+01	3.54E+01
F5	2.37E+03	9.98E+02	1.33E+03	1.11E+04	6.83E+06	1.29E+03	1.05E+03	1.17E+03	3.36E+05
F6	2.52E+02	2.32E+02	2.51E+02	3.19E+03	1.62E+03	5.16E+02	2.34E+02	1.91E+03	5.63E+03
F7	2.06E-02	4.41E-01	7.11E-02	7.66E-02	1.23E-01	1.19E+01	8.69E-02	4.21E+01	3.24E+01
F8	-3.89E+04	-4.12E+04	-3.86E+04	-3.89E+04	-4.09E+04	-3.78E+04	-3.88E+04	-3.78E+04	-3.95E+04
F9	4.67E+00	4.88E+02	2.88E+02	4.66E+01	2.26E+02	5.46E+00	1.68E+02	7.81E+01	5.52E+00
F10	2.48E-01	4.00E-01	1.77E-01	7.17E-01	4.55E-01	3.22E-01	1.73E-01	8.70E-03	1.06E+00
F11	5.46E-02	1.14E-01	5.40E+01	1.92E+00	2.79E+01	9.49E-02	7.72E+01	2.88E+00	1.89E+00
F12	1.22E+00	5.38E+04	2.67E+00	1.08E+00	1.04E+00	1.68E+00	1.06E+00	1.61E+05	1.23E+00
F13	1.68E+02	4.96E+02	1.71E+02	1.15E+02	4.80E+02	1.06E+02	2.15E+03	3.07E+06	1.14E+04
Total:	1.11E+06	1.09E+29	1.14E+79	6.50E+142	3.41E+131	1.79E+161	1.58E+10	3.99E+67	1.60E+06
Rank:	1	4	6	8	7	9	3	5	2

Article Information Form

Funding

The author (s) has no received any financial support for the research, authorship or publication of this study.

The Declaration of Conflict of Interest/ Common Interest

No conflict of interest or common interest has been declared by the authors.

The Declaration of Ethics Committee Approval

This study does not require ethics committee permission or any special permission.

The Declaration of Research and Publication Ethics

The authors of the paper declare that they comply with the scientific, ethical and quotation rules of SAUJS in all processes of the paper and that they do not make any falsification on the data collected. In addition, they declare that Sakarya University Journal of Science and its editorial board have no responsibility for any ethical violations that may be encountered, and that this study has not been evaluated in any academic publication environment other than Sakarya University Journal of Science.

References

[1] L. Abualigah, D. Yousri, M. Abd Elaziz, A. A. Ewees, M. A. Al-Qaness, A. H. Gandomi, “Aquila optimizer: a novel meta-heuristic optimization algorithm,” Computers & Industrial Engineering, vol. 157, pp. 1-37, 2021.

[2] C. L. Hwang, A. S. M. Masud, “Multiple Objective Decision Making— Methods and Applications: A State-of-the-Art Survey,” Berlin, Germany, Springer, 2012.

[3] P. Agrawal, H. F. Abutarboush, T. Ganesh, A. W. Mohamed, ‘‘Metaheuristic algorithms on feature selection: A survey of one decade of research (2009–2019),’’ IEEE Access, vol. 9, pp. 26766–26791, 2021.

[4] B. Abdollahzadeh, F. S. Gharehchopogh, N. Khodadadi, S. Mirjalili, “Mountain gazelle optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems,” Advances in Engineering Software, vol. 174, no. 103282, 2022.

[5] J. O. Agushaka, A. E. Ezugwu, L. Abualigah, “Gazelle optimization algorithm: a novel nature-inspired metaheuristic optimizer," Neural Computing and Applications, vol. *35*(5), pp. 4099-4131, 2023.

[6] A. A., Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, “Harris hawks optimization: Algorithm and applications,” Future Generation Computer Systems, vol. 97, pp. 849–872, 2019.

[7] S. Li, H. Chen, M. Wang, A. A. Heidari, S. Mirjalili, “Slime mould algorithm: A new method for stochastic optimization,” Future Generation Computer Systems, vol. 111, pp. 300–323, 2020.

[8] S. Kaur, L. K. Awasthi, A. L. Sangal, G. Dhiman, “Tunicate swarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization,” Engineering Applications of Artificial Intelligence, vol. 90, pp. 103541, 2020.

[9] A. Faramarzi, M. Heidarinejad, B. Stephens, S. Mirjalili, “Equilibrium optimizer: A novel optimization algorithm,” Knowledge-Based Systems, Vol. 191, pp. 105190, 2020.

[10] L. Abualigah, A. Diabat, S. Mirjalili, M. A. Elaziz, A. H. Gandomi, “The arithmetic optimization algorithm,” Computer Methods Applied Mechanics and Engineering, vol. 376, pp. 113609, 2021.

[11] A. Faramarzi, M. Heidarinejad, S. Mirjalili, A. H. Gandomi, “Marine predators algorithm: A nature-inspired metaheuristic,” Expert Systems with Applications, vol. 152, pp. 113377, 2020.

[12] E. Trojovská, M. Dehghani, P. Trojovský, “Zebra optimization algorithm: A new bio-inspired optimization algorithm for solving optimization algorithm,” IEEE Access, vol. 10, pp. 49445-49473, 2022.

[13] H. Jia, H. Rao, C. Wen, S. Mirjalili, “Crayfish optimization algorithm,” Artificial Intelligence Review, pp. 1-61, 2023.

[14] M. Noroozi, H. Mohammadi, E. Efatinasab, A. Lashgari, M. Eslami, B. Khan, “Golden search optimization algorithm,” IEEE Access, vol. 10, pp. 37515-37532, 2022.

[15] E. Baş, E. Ülker, “Comparison between SSA and SSO algorithm inspired in the behavior of the social spider for constrained optimization,” Artificial Intelligence Review, vol. 54(7), pp. 5583-5631, 2021.

[16] E. Baş, A. İhsan, “Performance analysis and comparison of gray wolf optimization and Krill herd optimization algorithm,” Pamukkale University Journal of Engineering Sciences, vol. 1000(1000), pp. 0-0, 2023.

[17] G. A. Grau, F. R. Walther, “Mountain gazelle agonistic behaviour,” Animal Behavior, vol. 24 (3), pp. 626–36, 1976.

[18] S. Omondi, “Gazelle Facts - Animals of the World. Retrieved from WorldAtlas / Environment, (2017, August 1). [Online]. Available: https://www. worldatlas.com/articles/gazelle-facts-animals-of-the-world.html.

[19] A. Einstein, “Investigations on the Theory of the Brownian Movement Courier Corporation,” US, 1956.

[20] R. N. Mantegna, “Fast, accurate algorithm for numerical simulation of Levy stable stochastic processes,” Physical Review E, vol. 49(5), pp. 4677, 1994.

[21] R. Rajabioun, “Cuckoo optimization algorithm,” Applied soft computing, vol. 11(8), pp. 5508-5518, 2011.

[22] A. Ahrari, M. R. Saadatmand, M. Shariat-Panahi, A. A. Atai, “On the limitations of classical benchmark functions for evaluating robustness of evolutionary algorithms,” Applied Mathematics and Computation, vol. 215(9), pp. 3222-3229, 2010.

[23] F. Wilcoxon, ‘‘Individual comparisons by ranking methods,’’ in Break throughs in Statistics, New York, NY, USA: Springer, 1992, pp. 196–202.