Neural Computing and Applications (2023) 35:4099–4131
https://doi.org/10.1007/s00521-022-07854-6

ORIGINAL ARTICLE

# Gazelle optimization algorithm: a novel nature-inspired metaheuristic optimizer

Jeffrey O. Agushaka[1,2] · Absalom E. Ezugwu[1] · Laith Abualigah[3,4,5]

Received: 20 May 2022 / Accepted: 16 September 2022 / Published online: 20 October 2022


## Abstract

This study proposes a novel population-based metaheuristic algorithm called the Gazelle Optimization Algorithm (GOA), inspired by the gazelles' survival ability in their predator-dominated environment. Every day, the gazelle knows that if it does not outrun and outmaneuver its predators, it becomes meat for the day, and to survive, the gazelles have to escape from their predators consistently. This information is vital to proposing a new metaheuristic algorithm that uses the gazelle's survival abilities to solve real-world optimization problems. The exploitation phase of the algorithm simulates the gazelles grazing peacefully in the absence of the predator or while the predator is stalking it. The GOA goes into the exploration phase once a predator is spotted. The exploration phase consists of the gazelle outrunning and outmaneuvering the predator to a safe haven. These two phases are iteratively repeated, subject to the termination criteria, and finding optimal solutions to the optimization problems. The robustness and efficiency of the developed algorithm as an optimization tool were tested using benchmark optimization test functions and selected engineering design problems (fifteen classical, ten composited functions, and four mechanical engineering design problems). The results of the GOA are compared with nine other state-of-the-art algorithms. The simulation results obtained confirm the superiority and competitiveness of the GOA algorithm over nine state-of-the-art algorithms available in the literature. Also, the standard statistical analysis test carried out on the results further confirmed the ability of GOA to find solutions to the selected optimization problems. It also showed that GOA performed better or, in some cases, was very competitive with some state-of-the-art algorithms. Also, the results show that GOA is a potent tool for optimization that can be adapted to solve problems in different optimization domains.

**Keywords** Gazelle optimization algorithm · GOA · Metaheuristics · Optimization problems · Industrial engineering design problems · Nature-inspired · Gazelle · Population-based

## 1 Introduction

Optimization problems are found in different spheres of human endeavors, such as engineering, medicine, computer science, and the production chain. Given some constraints, these problems are solved by either maximizing or minimizing the objective function to obtain an optimal solution. This is not an easy endeavor because these real-world problems are complex, nonlinear, multimodal, and gradient difficult. Metaheuristic algorithms have found application in solving optimization problems, among which are nature-inspired metaheuristic algorithms which are popular nowadays among researchers. These successes have been attributed to the manner these algorithms solve the problems [20]. The nature-inspired metaheuristic algorithms

✉ Absalom E. Ezugwu
Ezugwua@ukzn.ac.za

✉ Laith Abualigah
aligah.2020@gmail.com; aligah@ammanu.edu.jo

[1] School of Mathematics, Statistics, and Computer Science, University of KwaZulu-Natal, King Edward Road, Pietermaritzburg 3201, KwaZulu-Natal, South Africa

[2] Department of Computer Science, Federal University of Lafia, Lafia 950101, Nigeria

[3] Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman 19328, Jordan

[4] Faculty of Information Technology, Middle East University, Amman 11831, Jordan

[5] School of Computer Sciences, Universiti Sains Malaysia, Pulau Pinang 11800, Malaysia

mimic natural human and animal behavior and physical phenomena [24]. These algorithms have mimicked hunting, foraging, et al. mating, mutation, survival, swarming, and nesting to solve optimization problems [7, 21].

Nature-inspired metaheuristic algorithms use the evolutionary or exploratory approach to find optimal solutions to solve these problems [23]. This has led to a taxonomy of metaheuristic algorithms based on evolutionary algorithms [44], swarm intelligence algorithms [66], physical algorithms [50], and algorithms based on human behavior [38]. So many other taxonomies exist in the literature, a comprehensive consensus classification for metaheuristic algorithms has not yet been introduced. A contributing factor is the rate at which new algorithms are being proposed. The application of these metaheuristic algorithms and other computation intelligence methods in disease classification, clustering, and many others is well documented in the literature [4, 8].

Interestingly, there has been tremendous growth in the number of proposed metaheuristic algorithms that draw inspiration from nature within the last two decades. The inventors or developers of these algorithms often claim some novel and high-performing optimization process. Unfortunately, these claims are debatable because it is difficult to find a nexus between the optimization process of quite a few algorithms and natural phenomena or systems. Moreover, these algorithms have not been used to solve many complex real-world optimization problems since their first appearance in literature [6, 60]. However, as the No Free Lunch theorem observed, there is no guarantee that these proposed or existing metaheuristic algorithms would find optimal solutions to all or many real-world optimization problems. Hence, the need to develop new or hybrid methods to solve a specific problem or a range of problems optimally. It is also a testament that a specific optimization method can only solve certain problems efficiently but not all other problems.

Minimizing the values of design parameters and the overall design cost of the speed reducer design problem (SRD), pressure vessel design problem (PVD), compression spring design problem (CSD), and welded beam design problem (WBD) is a daunting task in the engineering domain. Many researchers have proposed novel or hybrid methods to find optimal solutions to these problems [15, 27, 29, 46, 51, 56, 58]. The is still the provision to find a better solution than the existing ones by developing a more robust and efficient algorithm. This motivated our work to develop a novel population-based algorithm inspired by the activities of the gazelles in the wild.

Although, studies have shown that quite a number of the proposed metaheuristic optimization methods are generally inspired by the foraging and hunting behaviors of animals in their natural habitat [25], and more so despite the interesting survival life cycle of some animals in the wild such as the gazelle, there is no modeling that conveys the attractive adaptive survival strategy of the gazelle in nature as an optimization process. Therefore, this study presents a new metaheuristic algorithm that models the survival ability of the Gazelles being among the top prey for predators in their natural habitat. The study modeled the ability of the gazelle to escape top predators by outrunning and outmaneuvering the predators in their environment. The proposed method has a professional mathematical model that simulates the Gazelle animal's intellectual movements in nature. The proposed algorithm is used to minimize the values of design parameters and hence the overall design cost of the speed reducer design problem (SRD), pressure vessel design problem (PVD), compression spring design problem (CSD), and welded beam design problem (WBD).

The technical contribution of this research can be expressed as follows:

- The novelty of this study is in the design of the new population-based metaheuristic algorithm called the GOA, which simulates the survival abilities of the gazelles in their natural habitat.
- The inspiration of GOA is the ability of the gazelles to spot and outrun or outmaneuver the predators. The survival adaptation is divided into two phases which correspond to the exploration and exploitation phases.
- The exploitation phase of the algorithm simulates the gazelles grazing peacefully in the absence of the predator or while the predator is stalking it. The GOA goes into the exploration phase once a predator is spotted, it consists of the gazelle outrunning and outmaneuvering the predator to a haven. These phases are repeated subject to termination criteria to find optimal solutions to optimization problems.
- The various stages of GOA are described and mathematically modeled.
- The efficacy and robustness of the GOA are evaluated by solving 25 benchmark functions (15 unimodal, high-dimensional multimodal, fixed-dimensional multimodal classical functions, and 10 composite functions defined in CEC2020).
- Also, the performance of GOA in solving real-world problems is evaluated using four engineering design problems.
- The optimization results obtained from GOA are compared with nine state-of-the-art algorithms to analyze the proposed algorithm's capability.

This article is organized as follows: In Sect. 2, the background of the research is discussed. The Gazelle Optimization Algorithm (GOA) is presented in Sect. 3. Section 4 presents the experimental design. A detailed

discussion of results and findings is presented in Sect. 5. Finally, the conclusion and future work is presented in Sect. 6.

## 2 Preliminaries

A detailed taxonomy of metaheuristic algorithms can be found in [24]. However, an abridged version of the taxonomy is given in Fig. 1 to enhance our discussion in this section. Specifically, Fig. 1 gives an abridged taxonomy of algorithms reviewed in this study. The evolutionary algorithms mimic Darwin's laws of evolution and competence to solve optimization problems. The search agents' position vector is randomly generated and evolved during optimization using different updating rules uniquely defined for every algorithm. Popular in this category are the genetic algorithms (GA) [28]; others include the differential evolution (DE) [59], the artificial algae algorithm (AAA) [61], and the evolution strategy (ES) [54].

The second category is based on physical methods that typically occur in physics and chemistry. These algorithms mimic the universe's physical rules. Some examples of the physical-based methods are the simulated annealing (SA) [36], the water wave optimization (WWO) algorithm [31], gravitational search algorithm (GSA) [14, 52], and the artificial chemical reaction optimization algorithm (ACROA) [11].

The third category is the swarm-based methods, which consist of algorithms whose optimization process imitates animals' social behavior or activities in groups to find the optimal solution. In this type of algorithm, the population shares intelligence by one solution searching in parallel with the help of others to find the optimal solution. The birds, insects, and some arthropods exhibit these behaviors and form a subcategory for this category. Popular in this category are particle swarm optimization (PSO) [35], artificial bee colony (ABC) [9, 10], and ant colony optimization (ACO) [17].

Insects like bees, fireflies, butterflies, and water striders are known to exhibit swarm intelligence. Studies have shown that they have social and structured behaviors and live together in a swarm or group. An insect represents a solution, and each insect collaborates to search for a solution to the problem. Popular in this category are the artificial bee colony (ABC) [9, 10], the firefly algorithm (FA) [22], the butterfly optimization algorithm (BOA) [12], the Water strider algorithm (WSA) [32].

Metaheuristic algorithm such as teaching–learning-based optimization (TLBO) [55] mimics human behavior. Others include the imperialist competitive algorithm (ICA) [13], and rich/poor optimization (RPO) [45]. The TLBO mimics the behavior of the different actors in the learning

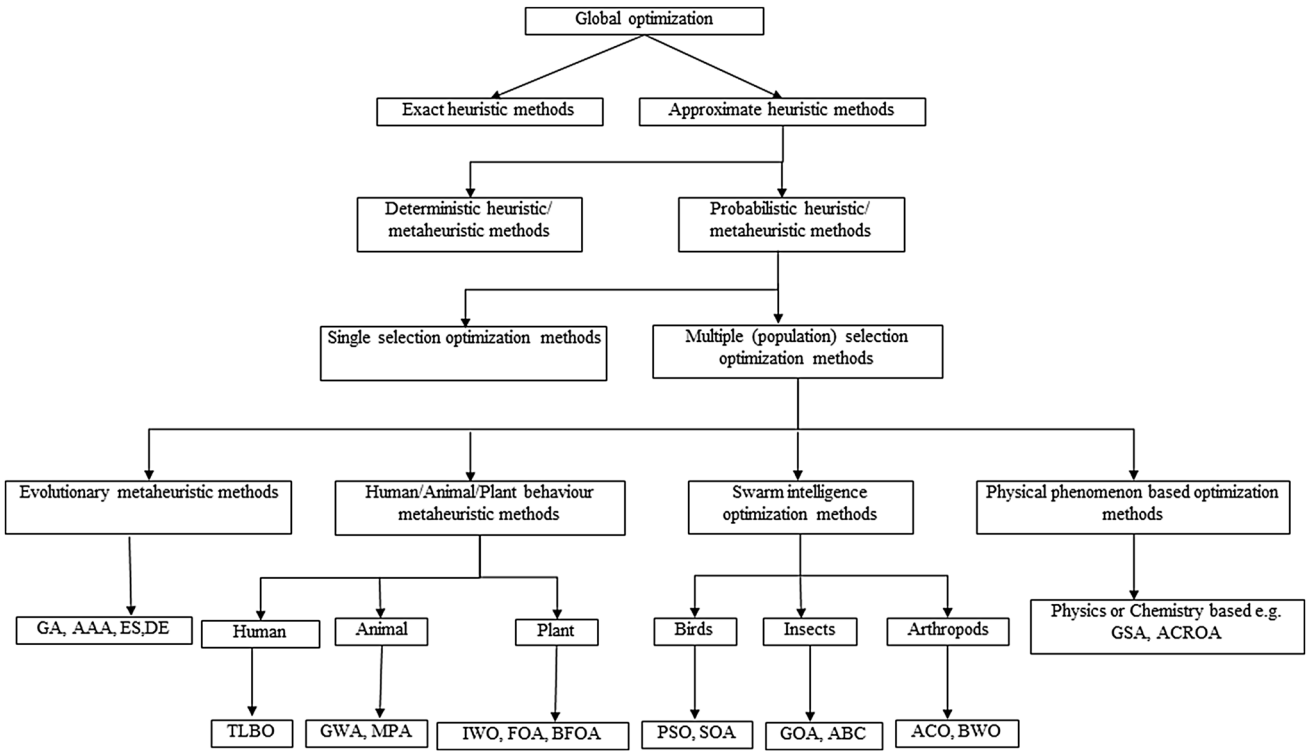


**Fig. 1** A brief taxonomy of metaheuristic optimization methods

environment, whereas the RPO mimics the behavior of the poor trying to reach the level of the rich people, thereby eliminating the class gap. Some metaheuristic algorithms mimic behaviors such as plant growth. The invasive weed optimization algorithm (IWO) [64] and a forest optimization algorithm (FOA) [63] are examples of this subcategory.

New metaheuristic algorithms are being proposed at a high rate; most have a different source of inspiration, while others improve existing ones. The arithmetic optimization algorithm (AOA) was proposed by [3] and uses the distributive power of simple arithmetic operators to find an optimal solution for optimization problems. Using advanced arithmetic operators like the natural logarithm and exponential operators, [5] proposed the nAOA, which enhanced the exploratory and exploitation ability of the original AOA. Similarly, the krill herd algorithm (KH) was improved using the low discrepancy sequences for the initial population [4, 8]. A novel optimization method called ebola optimization search algorithm (EOSA), which mimics how the Ebola virus spreads, was proposed by [49]. Others include [33, 68].

The proposed novel metaheuristic algorithms have been successfully applied to find solutions that are optimal or near-optimal for problems in different domains, including the engineering field [1, 2, 67, 70]. These authors claim novelty for their algorithms, citing a new source of inspiration. However, there is still room for improvement in the quality of results obtained by these algorithms. It is believed that a more efficient and robust optimizer would be able to find better solutions for complex optimization problems.

Hybridization is another approach to developing a more robust optimizer. In this approach, the authors would seek to leverage the strong capability of the specific optimizers to perform the task of either exploration or exploitation mechanism. These vital characteristics are combined with another optimizer to obtain a resultant optimization method with enhanced exploration and exploitation features. Moreover, the general goal in most cases is to improve or hybridize the existing methods to solve challenging optimization problems. Several existing algorithms have been shown to have a limited approach to solving complicated real-world optimization problems. The hybrid algorithms offer solutions to most complex problems, and their claims are often tested and validated using real-world problems [7, 34, 37, 62, 69]

## 3 The gazelle optimization algorithm (GOA)

In this section, the gazelle optimization algorithm (GOA) is presented, and the proposed steps for optimization are formulated.

### 3.1 Motivation

Minimizing the values of design parameters and the overall design cost of engineering design problems such as the speed reducer design problem (SRD), pressure vessel design problem (PVD), compression spring design problem (CSD), and welded beam design problem (WBD) is a daunting task. As mentioned earlier, many researchers have proposed novel or hybrid methods to find optimal solutions to these problems. However, it is believed that room still exists to find a better solution than the existing ones by developing a more robust and efficient algorithm. Therefore, this motivated our proposed work to develop a novel population-based algorithm inspired by the activities of the gazelles.

Although studies have shown that quite a number of the proposed metaheuristic optimization methods are generally inspired by the foraging and hunting behaviors of animals in their natural habitat and more so despite the interesting survival life cycle of some animals in the wild such as the gazelle, there is no modeling that conveys the attractive adaptive survival strategy of the gazelle in nature as an optimization process. The gazelle knows that if it does not outrun and outmaneuver its predators, it becomes meat for the day. The gazelles are down in the food chain and one of the most hunted prey in their habitat. They are not considered endangered; this means they are doing something right. One of those right attributes inherent in the gazelles is their escape from predators. Studies show that the predators are only successful 34% most of the time. This paper employs this information to develop a new metaheuristic algorithm that solves real-world problems using the gazelle's survival abilities.

### 3.2 Inspiration

The gazelles inhabit the drylands covering parts of Asia, including China, stretching the Arabian Peninsula, and parts of the Sahara desert in northern Africa. They are also found in the sub-Saharan Sahel and northeast Africa, stretching from the Horn of Africa to Tanzania. The gazelles are one of the common prey for most predators. The gazelles belong to the genus Gazella family [19]. About 19 different gazelles exist globally, ranging from small gazelles like Thomson's and Speke's gazelle to the large gazelle-like the Dama gazelle [48]. Gazelles are light

and swift and have a strong sense of hearing, sight, and smell. These adaptive characteristics compensate for their apparent habitual vulnerabilities by allowing them to run away from their predators. The gazelles and their unusual behaviors can be observed in nature.

Gazelles are classified as herbivorous animals and feed on only plants like leaves, grasses, shoots, and other plant-based food. Like most primary consumers, they adapt to living in groups for security and social reasons. They socialize within the group, and up to 700 members can be found in a group. A large number of gazelles in a group helps the group achieve herd security. Sometimes, the social structure of the gazelle group can be based on sexes, where the females live in a smaller group with their fawns; this is common with Thomson's gazelles. On the other hand, the bachelor's herd consists only of males, who help provide for and defend the group. The gazelles have a birthrate of one or two, the frequency of birth is twice per year, with a gestation period lasting up to 6 months. It is more convenient for the gazelles to breed during the rainy season because of abundant food and water for all the group, including the newborns.

The gazelles occupy level 2 in the food chain; they are the primary prey for many predators. Typical predators of the gazelles are humans, cheetah, Asiatic and black-backed jackal, spotted hyena, wild dog, leopard, and lion [26]. In the face of danger, the gazelles warn each other by flicking their tail, stomping their feet, or leaping in the air. The act of leaping with all four feet up to 2 m in the air is called "stotting." Although the exact cause of stotting is unknown, this behavior is exhibited when nervous or otherwise excited. Also, for predators such as cheetahs and lions, the gazelle runs at exceptional speeds, reaching a top speed of 100 km/hr. A gazelle can outrun and outmaneuver the fastest predator on land (cheetah). The success of most predators depends on how well they stalk the gazelle stealthily; without the element of surprise, the gazelle most times would outrun and outmaneuver the predator at high speed.

A study of Mongolian Gazelles discovered that human activities' estimated daily hazard rate far outweighs that of the hazard rate due to natural predation. They are causing a mortality rate of 30% higher [47]. The authors also posited that though the gazelles are not endangered species, they have an annual survival rate of 0.66 which can be translated to mean predators are only successful in 0.34 situations. The following points express the gazelles' survival methods, which would be used to model the proposed algorithm.

- The most spectacular aspects are grazing and running from predators.
- The grazing aspect in the absence of predators can be used for exploitation
  - o Predators stalk gazelles while grazing
  - p The gazelle use stotting to, among others, spot predators
  - q Height of 2 m
- The ability to outrun spotted predators to a haven can be used for exploration.
  - o It cannot outrun the fastest predator but outmaneuvers it
  - p 88km/hr

### 3.3 Population initialization

The GOA is a population-based optimization algorithm that uses randomly initialized gazelles (X) as search agents. The search agents are defined as a $n \times d$ matrix of candidate solutions as defined in Eq. (1). The GOA uses the problem's constraint of upper bound (UB) and lower bound (LB) to stochastically define the range of values that the population vector can take.

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

where X is the matrix of the position vector of the candidate population, each position vector is stochastically generated using Eq. (2), $x_{i,j}$ is the randomly generated vector position of the $i^{th}$ population in the $j^{th}$ dimension, n represents the number of gazelles, and d is the defined search space (dimension) of the optimization problem.

$$x_{i,j} = \text{rand} \times \left(\text{UB}_j - \text{LB}_j\right) + \text{LB}_j \quad (2)$$

where *rand* is a random number, $UB_j$ *and* $LB_j$ are upper and lower bound of the problem, respectively. In every iteration, each $x_{i,j}$ produces a candidate solution. The minimum solution is taken to be the best-obtained solution so far. It is said that the strongest or fittest gazelles in nature are more talented in spotting, informing others of the treats, and running from predators. Thus, the best-obtained solution so far is nominated as a top gazelle to construct an Elite $n \times d$ matrix (Eq. 3). This matrix is used for searching and finding the next step for the gazelles.

$$\text{Elite} = \begin{bmatrix} x'_{1,1} & x'_{1,2} & \cdots & x'_{1,d-1} & x'_{1,d} \\ x'_{2,1} & x'_{2,2} & \cdots & x'_{2,d-1} & x'_{2,d} \\ \vdots & \vdots & x'_{i,j} & \vdots & \vdots \\ x'_{n,1} & x'_{n,2} & \cdots & x'_{n,d-1} & x'_{n,d} \end{bmatrix} \quad (3)$$

where $x'_{i,j}$ represents the position vector of the top gazelle. The GOA considers both the predator and gazelles as search agents. Because by the time a predator is spotted stalking the gazelles, they both run in the same direction towards the haven, and when the gazelles escape, the predator will have also explored the search space. The Elite will be updated at the end of each iteration if the better gazelle substitutes the top gazelle.

#### 3.3.1 The Brownian motion

A random motion in which the displacement follows the Normal (Gaussian) probability distribution function with specific mean and unit variance of $\mu = 0$ and $\sigma^2 = 1$, respectively. At point $x$, the standard Brownian motion [18] is defined in Eq. 4.

$$f_B(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right) \tag{4}$$

#### 3.3.2 The Lévy flight

The Lévy flight performs a random walk using the Lévy distribution (power-law tail) given in Eq. 5 [30].

$$L(x_j) \approx |x_j|^{1-\alpha} \tag{5}$$

where $x_j$ denotes the flight distance, and $\alpha = (1, 2]$ represents the power-law exponent. Equation 6 denotes the Lévy stable process as an integral [39].

$$f_L(x;\alpha,\gamma) = \frac{1}{\pi}\int_0^\infty \exp(-\gamma q^\alpha)\cos(qx)\delta q \tag{6}$$

where $\alpha$ is the distribution index that controls everything about the motion, and $\gamma$ denotes the scale unit. Our work used an algorithm that generates a stable Lévy motion proposed by [39]. The algorithm uses $\alpha$ within the range of 0.3–1.99, and it is defined as in Eq. 7.

$$\text{Levy}(\alpha) = 0.05 \times \frac{x}{|y|^{\frac{1}{\alpha}}} \tag{7}$$

where $\alpha, x$ and $y$ are defined as follows: $x = Normal(0, \sigma_x^2)$ and $y = Normal\left(0, \sigma_y^2\right)$

$$\sigma_x = \left[\frac{\Gamma(1+\alpha)sin\left(\frac{\pi\alpha}{2}\right)}{\Gamma\left(\frac{(1+\alpha)}{2}\right)\alpha 2^{\frac{(\alpha-1)}{2}}}\right]^{1/\alpha}, \sigma_y = 1, \text{ and } \alpha = 1.5$$

### 3.4 Modeling the GOA

The proposed GOA algorithm simulates the survival behavior of the gazelles. The optimization process comprises grazing without a predator and running from a spotted predator to a haven. Hence, the proposed GOA algorithm optimization process consists of two phases.

#### 3.4.1 Exploitation

This phase assumes the gazelles are grazing peacefully without a predator or while the predator is stalking the gazelles. In this phase, the Brownian motion characterized by uniform and controlled steps was used to effectively cover neighborhood areas of the domain. The gazelles are assumed to move in Brownian motion while grazing, as depicted in Fig. 2.

The mathematical model of this behavior is as shown in Eq. 8.

$$\begin{aligned}\text{gazelle}_{i+1} = {} & \text{gazelle}_i + s.R * .R_B \\ & * .(\text{Elite}_i - R_B * .\text{gazelle}_i)\end{aligned} \tag{8}$$

where $gazelle_{i+1}$ is the solution of the next iteration, $gazelle_i$ is the solution at the current iteration, s denotes the grazing speed of the gazelles, $\overrightarrow{R}_B$ is a vector containing random numbers representing the Brownian motion, $R$ is a vector of uniform random numbers [0,1].

#### 3.4.2 Exploration

The exploration phase kicks off the moment a predator is sighted. The gazelles react to danger by flicking their tail, stomping their feet, or stotting up to 2 m in the air with all four feet; simulated by scaling the 2 m height to a number between 0 and 1. The Lévy flight is used for this algorithm phase; the process consists of taking tiny steps and occasionally long jumps. This approach has improved searchability in optimization literature [65]. The exploration phase is depicted in Fig. 3. The gazelle runs once the predator is spotted, and the predator chases. Both runs are characterized by a sudden change of direction, which is represented by the $\mu$. This study assumed this change of direction occurs in every iteration; when the iteration number is odd, the gazelle goes one way and the other way when the iteration is even. The gazelle reacts first, so we assumed it runs using Lévy flight. The predator reacts later, so the study assumed its take-off run using the Brownian motion before changing to the Lévy flight later. The mathematical model of the behavior of the gazelle once it spots the predator is as shown in Eq. 9.

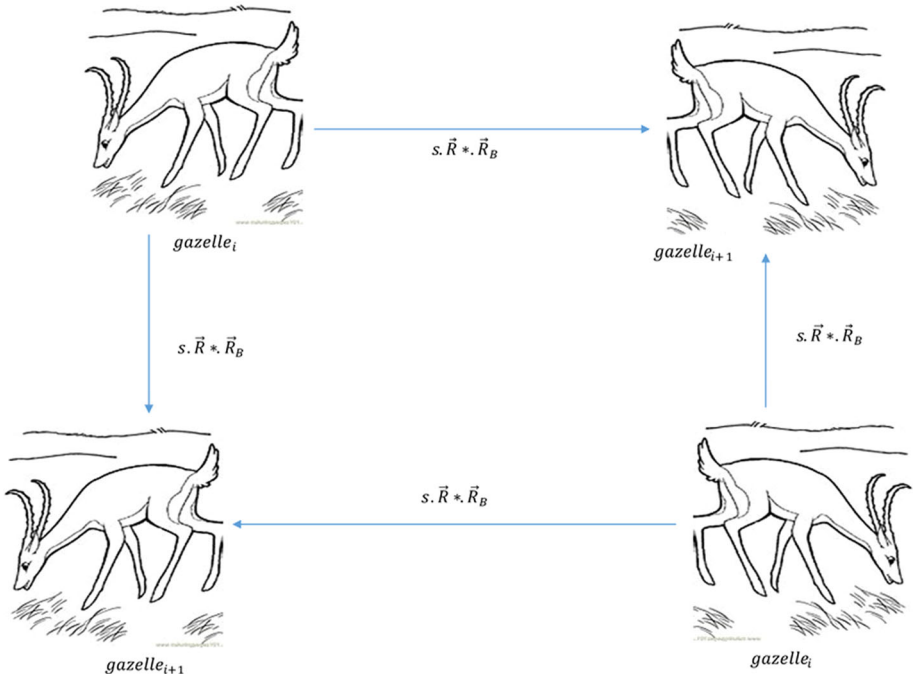


**Fig. 2** The grazing behavior of the gazelles signifies exploitation

$$\overrightarrow{\text{gazelle}_{i+1}} = \overrightarrow{\text{gazelle}_i} + S.\mu.\vec{R} * .\vec{R}_L * .\left(\overrightarrow{\text{Elite}_i} - \overrightarrow{R_L} * .\overrightarrow{\text{gazelle}_i}\right) \quad (9)$$

where $S$ is the top speed, the gazelle can reach, $\overrightarrow{R}_L$ denotes a vector of random numbers based on Lévy distributions. The mathematical model for the behavior of the predator chasing the gazelle is shown in Eq. 10.

$$\overrightarrow{\text{gazelle}_{i+1}} = \overrightarrow{\text{gazelle}_i} + S \cdot \mu \cdot \text{CF} * \cdot\vec{R}_B * \cdot\left(\overrightarrow{\text{Elite}_i} - \overrightarrow{R_L} * \cdot\overrightarrow{\text{gazelle}_i}\right) \quad (10)$$

where $CF = \left(1 - \frac{iter}{Max\_iter}\right)^{\left(2\frac{iter}{Max\_iter}\right)}$ denotes the cumulative effect of the predator.

In a study of Mongolian Gazelles, the authors also posited that though the gazelles are not endangered species, they have an annual survival rate of 0.66 which can be translated to mean predators are only successful in 0.34 situations [47]. PSRs is the predator success rates, the effect affects the ability of the gazelle to escape, which means the algorithm avoids being trapped in a local minimum. The PSRs effect is modeled as in Eq. 11.

$$\overrightarrow{\text{gazelle}_{i+1}} = \begin{cases} \overrightarrow{\text{gazelle}_i} + \text{CF}\left[\overrightarrow{\text{LB}} + \vec{R} * .\left(\overrightarrow{\text{UB}} - \overrightarrow{\text{LB}}\right)\right] * \cdot\vec{U} & \text{if } r \le \text{PSRs} \\ \overrightarrow{\text{gazelle}_i} + [\text{PSRs}(1-r) + r]\left(\overrightarrow{\text{gazell}_{r_1}} - \overrightarrow{\text{gazelle}_{r_2}}\right) & \text{else} \end{cases} \quad (11)$$

where $\overrightarrow{U}$ denotes a binary vector, which is constructed by generating a random number $r$ in [0,1] such that $\overrightarrow{U} = \begin{cases} 0, if\ r < 0.34 \\ 1, otherwise \end{cases}$. $r_1$ *and* $r_2$ are random indexes of the gazelle matrix.

### 3.5 Pseudocode for the proposed algorithm

The exploitation phase of the algorithm simulates the gazelles grazing peacefully in the absence of the predator or while the predator is stalking it. The GOA goes into the exploration phase once a predator is spotted. This phase consists of the gazelle outrunning and outmaneuvering the predator to a safe haven. The two steps are repeated subject to termination criteria to find optimal solutions to optimization problems. The implementation flow of these phases, as defined by their respective mathematical model, is shown in the pseudocode for the GOA given below.

**Algorithm 1:** Pseudocode of the GOA Optimizer

***begin***
*Initialize the algorithm parameters*
*s= [0,1]*
*μ= [-1,1]*
*S= 88kmph*
*PSRs=0.34*
*R and r are random numbers [0,1]*
*Initialize the gazelle populations (search agents)*
***While*** iter<max_iter
    *Calculate the fitness of the gazelles*
    *top-gazelle=fittest gazelle*
    *Construct the Elite gazelle matrix*
    ***If*** *r<0.5*
        ***Exploitation:***
            *Update gazelles based on*
            $gazelle_{i+1} = gazelle_i + s.R *. R_B *. (Elite_i - R_B *. gazelle_i)$
    ***Else***
        ***Exploration:***
            ***If*** *mod(iter,2)==0*
                $\mu = -1$
            ***Else***
                $\mu = 1$
            ***If*** iter<size(gazelle,1)/2
                *For the gazelle populations (i = 1,…,n/2)*
                *Update gazelles based on*
                $gazelle_{i+1} = gazelle_i + S.\mu.R *. R_L *. (Elite_i - R_L *. gazelle_i)$
            ***Else***
                *For the predator populations (i = n/2,…,n)*
                *Update gazelles based on*
                $gazelle_{i+1} = gazelle_i + S.\mu.CF *. R_B *. (Elite_i - R_L *. gazelle_i)$
    ***End (if)***
    *Fitness update*
    *top-gazelle update*
    *Elite update*
    *Applying PSRs effect and update based on*

$$gazelle_{i+1} = \begin{cases} gazelle_i + CF[LB + R *. (UB - LB)] *. U & if\ r \le PSRs \\ gazelle_i + [PSRs(1-r) + r]\left(gazell_{r_1} - gazelle_{r_2}\right) & else \end{cases}$$

***End while***
***Return*** *top-gazelle from the population*
***End***

### 3.6 Computational complexity

The dominant operations in the GOA optimization process typically rely on solution initialization, calculating the fitness functions, and updating solutions. There is $n$ number of solutions, which corresponds to $O(n)$. The complexity of evolving the candidate solutions is $O(iter \times d) + O(CFE)$. The updating process relies on equations modeled to explore the best positions that guarantee the optimal solution and correspondingly update the rest of the solutions' positions towards the best. Therefore, the computational complexity of the whole optimization processes involved in GOA is $O(iter \times d \times n + CFE \times n)$

where the total iterations permitted is called *iter*, the dimensionality of the given problem is called $d$, and *CFE* is the cost of evaluating the respective functions. The detailed optimization process flowchart of GOA is shown in Fig. 4. The algorithm starts by initializing the parameters s, μ, S, PSRs, R, and r. Also, the size of the gazelle's population and the maximum number of iterations are initialized. The

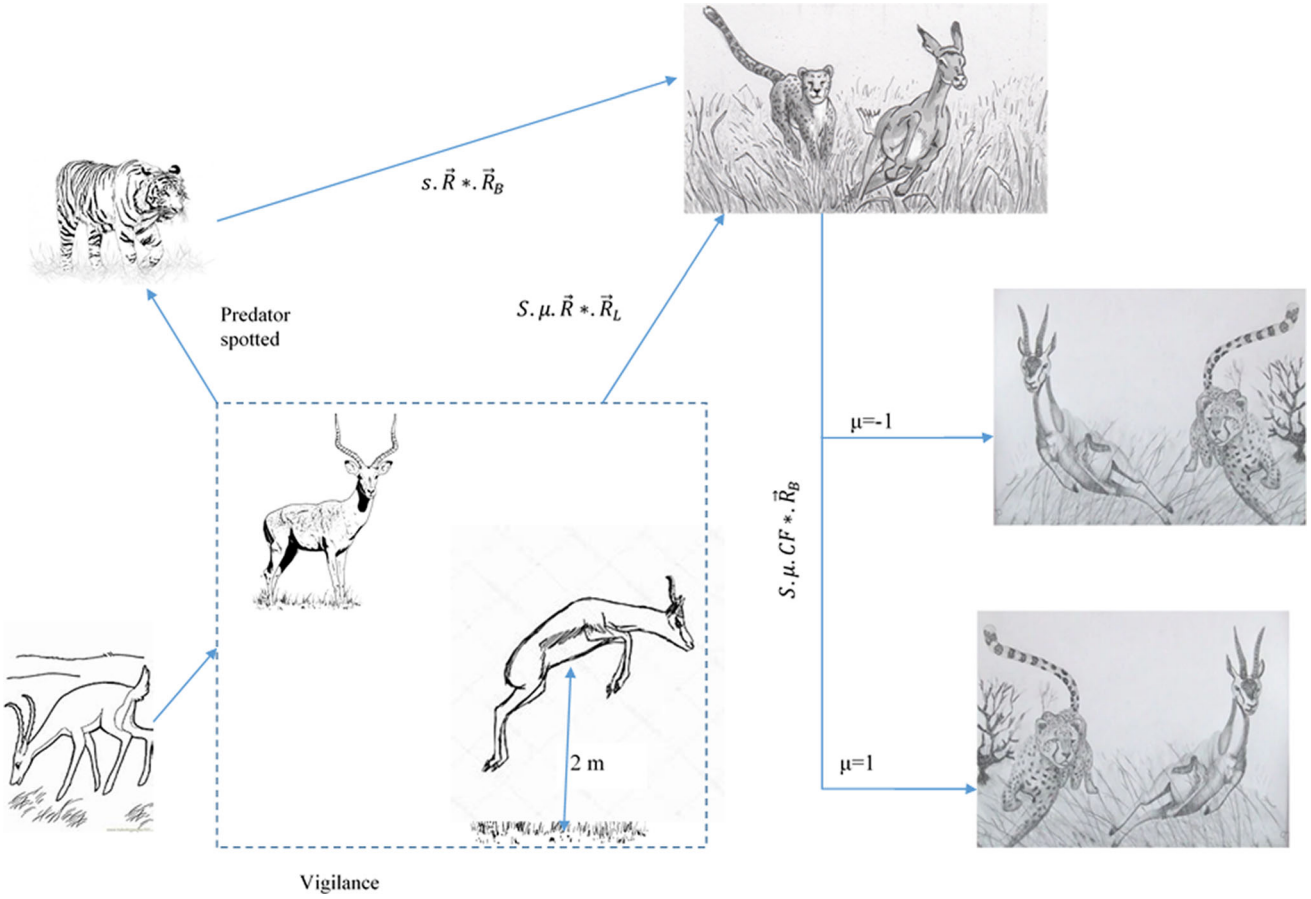


**Fig. 3** Gazelles escaping from the predator signifies exploration

GOA moves to generate the initial population stochastically, and then the process of evolving the initial population is repetitively carried out until a stop criterion is reached. The GOA evolves the initial population by either performing exploitation or exploration. The GOA is in exploitation mode until a predator is spotted (simulated by $r > 0.5$). In each case, the obtained solution is compared with the previously found solution; then, the global best-obtained solution is updated. This process continues until the stop criterion is reached. Once this happens, the global best-obtained result is returned. Finally, GOA terminates once the stop criterion is reached.

## 4 Experiment design

This section presents the design of experiments conducted to evaluate the performance of GOA. Fifteen (15) classical test functions, ten (10) CEC 2020 test functions, and four (4) selected optimization design problems in the engineering domain were used to evaluate the GOA. The results obtained were compared with that of the following algorithms:

- Advanced arithmetic optimization algorithm (nAOA) [5]
- Arithmetic optimization algorithm (AOA) [3]
- Constriction coefficient-based PSO and GSA (CPSOGSA) [53]
- Particle swarm optimization (PSO) [35]
- Biogeography-based optimization (BBO) [57]
- Differential evolution (DE) [59]
- Salp swarm algorithm (SSA) [42]
- Sine cosine algorithm (SCA) [41]
- Grey wolf optimizer (GWO) [43]

The original MATLAB source code (available online) of the nine (9) state-of-the-art algorithms were downloaded and used. The GOA was implemented from scratch using MATLAB R2020b. The MATLAB implementation of the selected engineering problem was also used to evaluate the algorithms. The hardware used consists of the Windows 10 operating system, Intel Core i7-7700@3.60 GHz processor, and 16 G RAM. The population size and the maximum number of iterations are sensitive parameters and need to be tuned. For this study, the population size is tuned to 50, and the maximum number of iterations is tuned to 1000.

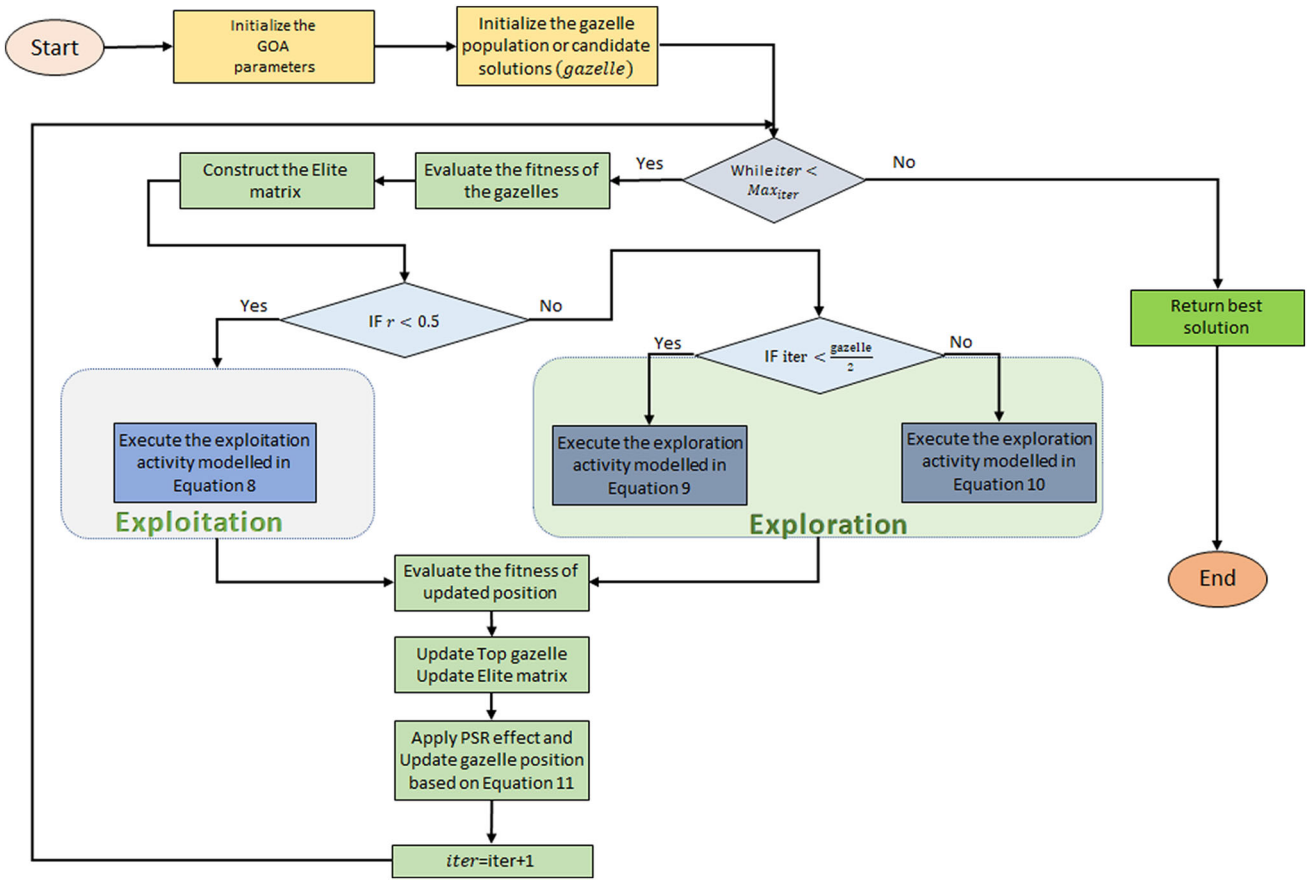


**Fig. 4** The flowchart of GOA

**Table 1** The individual algorithms control parameters settings

| Algorithm | Name of the parameter | Value of the parameter |
|---|---|---|
| AOA, nAOA | $\alpha$ | 5 |
| | $\mu$ | 0.05 |
| DE | Lower bound of scaling factor | 0.2 |
| | Upper bound of scaling factor | 0.8 |
| | PCR (crossover probability) | 0.8 |
| PSO | $C_1$, $C_2$ (personal and social constants) | 2 |
| | Wmax (maximum inertia weight) | 0.9 |
| | Wmin (minimum inertia weight) | 0.2 |
| CPSOGSA | $< p1, < f > 2$ (control parameters) | 2.05 |
| BBO | nKeep (number of habitats retained after every generation) | 0.2 |
| | Pmutation (mutation probability) | 0.9 |
| GWO | a (area vector) | [0,2] |
| | $r_{1f}$ $r_2$ (random vectors) | [0,1] |
| SCA | a (constant) | 2 |
| SSA | $c_2$, $c_3$ (random numbers) | [0,1] |

The stop criterium is the maximum number of iterations. The number of independent runs for each algorithm is set at 30. The implementation source codes for the respective optimization algorithm are also publicly available from the respective references. The control parameters of each algorithm used in this work are defined in the original article proposing the respective algorithms and given in Table 1.

**Table 2** Classical test functions

| ID | Type | Function | Dimension | Bounds | Global |
|---|---|---|---|---|---|
| F1 | Unimodal | $f(x)=\sum_{i=1}^{n}x_i^2$ | 30 | [− 100,100] | 0 |
| F2 | Unimodal | $f(x)=\sum_{i=0}^{n}\lvert x_i\rvert+\prod_{i=0}^{n}\lvert x_i\rvert$ | 30 | [− 10,10] | 0 |
| F3 | Unimodal | $f(x)=\sum_{i=1}^{d}\left(\sum_{j=1}^{i}x_j\right)^2$ | 30 | [− 100,100] | 0 |
| F4 | Unimodal | $f(x)=\max_i\{\lvert x_i\rvert,1\le i\le n\}$ | 30 | [− 100,100] | 0 |
| F5 | Unimodal | $f(x)=\sum_{i=1}^{n-1}\left[100(x_i-x_{i+1})^2+(1-x_i)^2\right]$ | 30 | [− 30,30] | 0 |
| F6 | Unimodal | $f(x)=\sum_{i=1}^{n}([x_i-0.5])^2$ | 30 | [− 100,100] | 0 |
| F7 | Unimodal | $f(x)=\sum_{i=0}^{n}\mathrm{i}x_1^4+\mathrm{rand}[0,1)$ | 30 | [− 128,128] | 0 |
| F8 | Multimodal | $f(x)=\sum_{i=1}^{n}-x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)$ | 30 | [− 500,500] | $-418.9829\times n$ |
| F9 | Multimodal | $f(x)=10+\sum_{i=1}^{n}\left(x_i^2-10\cos(2\pi x_i)\right)$ | 30 | [− 5.12,5.12] | 0 |
| F10 | Multimodal | $f(x)=-a\exp\left(-0.02\sqrt{n^{-1}\sum_{i=1}^{n}x_i^2}\right)-\exp\left(n^{-1}\sum_{i=1}^{n}\cos(2\pi x_i)\right)+a+e, a=20$ | 30 | [− 32,32] | 0 |
| F11 | Multimodal | $f(X)=1+\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)$ | 30 | [− 600,600] | 0 |
| F12 | Multimodal | $f(x)=\frac{\pi}{n}\{10\sin(\pi y_i)\}+\sum_{i=1}^{n-1}(y_i-1)^2\left[1+10\sin^2(\pi y_{i+1})+\sum_{i=1}^{n}u(x_i,10,100,4)\right]$ Where $y_i=1+\frac{x_i+1}{4}, u(x_i,a,k,m)\begin{cases}K(x_i-a)^m & ifx_i>a\\ 0 & -a\le x_i\ge a\\ K(-x_i-a)^m & -a\le x_i\end{cases}$ | 30 | [− 50,50] | 0 |
| F13 | Multimodal | $f(x)=0.1\left(\sin^2(3\pi x_1)+\sum_{i=1}^{n}(x_i-1)^2\left[1+\sin^2(3\pi x_i+1)\right]+(x_n-1)^2 1+\sin^2(2\pi x_n)\right)+\sum_{i=1}^{n}u(x_i,5,100,4)$ | 30 | [− 50,50] | 0 |
| F14 | Fixed-dimension multimodal | $f(x)=\sum_{i=1}^{11}\left[a_i-\frac{x_1\left(b_i^2+b_i x_2\right)}{b_i^2+b_i x_3+x_4}\right]^2$ | 4 | [− 5,5] | 0.00030 |
| F15 | Fixed-dimension multimodal | $f(X)=\left[1+(x_1+x_2+1)^2\left(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2\right)\right]\times\left[30+(2x_1-3x_2)^2\left(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2\right)\right]$ | 2 | [− 2,2] | 3 |

**Table 3** The CEC 2020 functions

| Type | Number | Functions | $F_i^*$ |
|---|---|---|---|
| Unimodal function | F16 | Shifted and rotated bent cigar function (CEC 2017[4] F1) | 100 |
| Basic functions | F17 | Shifted and rotated Schwefel's function (CEC 2014[3] F11) | 1100 |
| | F18 | Shifted and rotated Lunacek bi-Rastrigin function (CEC 2017[4] F7) | 700 |
| | F19 | Expanded Rosenbrock's plus Griewangk's function (CEC2017[4]f19) | 1900 |
| Hybrid functions | F20 | Hybrid function 1 (N = 3) (CEC 2014[3] F17) | 1700 |
| | F21 | Hybrid function 2 (N = 4) (CEC 2017[4] F16) | 1600 |
| | F22 | Hybrid function 3 (N = 5) (CEC 2014[3] F21) | 2100 |
| Composition functions | F23 | Composition function 1 (N = 3) (CEC 2017[4] F22) | 2200 |
| | F24 | Composition function 2 (N = 4) (CEC 2017[4] F24) | 2400 |
| | F25 | Composition function 3 (N = 5) (CEC 2017[4] F25) | 2500 |

*Search range: $[-100,100]^D$

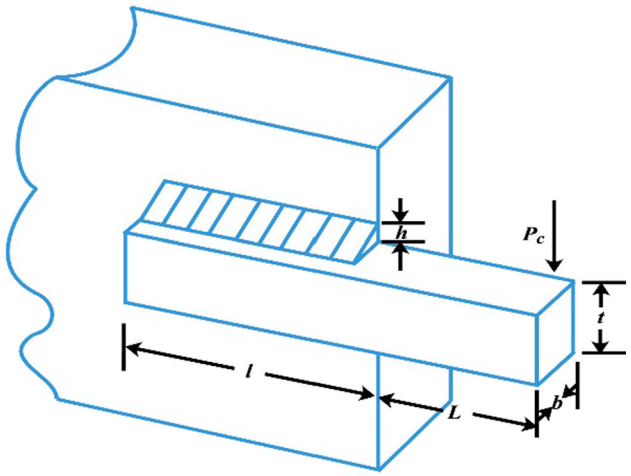


**Fig. 5** The welded beam design problem

## 4.1 Benchmark Functions

Also, the classical and composite benchmark functions used to test the robustness and efficiency of the proposed algorithm are presented in Tables 2 and 3, respectively. The benchmark functions consist of unimodal, multimodal, fixed-dimension multimodal, and composite functions. All the benchmark functions have different complexities in terms of the number and location of the global optimum solution. The unimodal functions are suitable for testing exploitation, while the multimodal functions are suitable for evaluating the exploration. The composite test functions shift the global optimum to random locations, making it challenging to find the global optimum.

## 4.2 Engineering problem

The goal of applying GOA to engineering design problems is to find the minimum cost and values of the corresponding design parameters. The four (4) problems considered for our experiments are discussed in the next section.

### 4.2.1 The welded beam design problem (WBD)

The welded beam design problem is formulated specifically to reduce the manufacturing cost of the design [16]. A simple illustration of the WBD is shown in Fig. 5. The WBD constraints are shear ($\tau$) and beam blending ($\theta$) stress, bar buckling load ($P_c$), beam end deflection ($\delta$), and side constraints defined as follows:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} h \\ l \\ t \\ b \end{bmatrix}$$

with the length ($l$), height ($t$), thickness ($b$) and weld thickness ($h$) of the bar

The WBD problem is mathematically formulated as defined in Eq. 12 [16]:

$$\begin{aligned} \text{Given } \vec{l} &= [l_1 l_2 l_3 l_4] = [hltb] = [x_1 x_2 x_3 x_4] \min f(\vec{l}) \\ &= 1.10471 l_1^2 l_2 + 0.04811 l_3 l_4 (14.0 + l_2) \end{aligned} \tag{12}$$

subject to

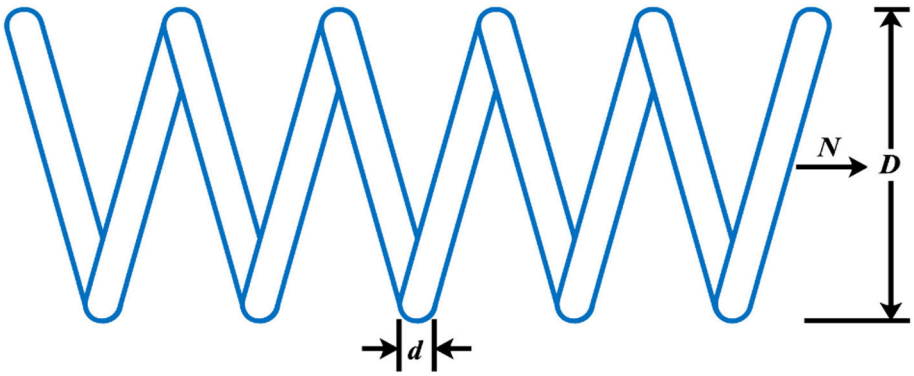


**Fig. 6** Compression spring design problem

$$s_1\left(\vec{l}\right) = \tau\left(\vec{l}\right) - \tau_{\max} \leq 0,$$
$$s_2\left(\vec{l}\right) = \sigma\left(\vec{l}\right) - \sigma_{\max} \leq 0,$$
$$s_3\left(\vec{l}\right) = l_1 - l_4 \leq 0,$$
$$s_4\left(\vec{l}\right) = 1.10471 l_1^2 + 0.04811 l_3 l_4 (14.0 + l_2) - 5.0 \leq 0,$$
$$s_5\left(\vec{l}\right) = 0.125 - l_1 \leq 0,$$
$$s_6\left(\vec{l}\right) = \delta\left(\vec{l}\right) - \delta_{\max} \leq 0,$$
$$s_7\left(\vec{l}\right) = P - P_c\left(\vec{l}\right) \leq 0,$$

The intervals for the design variables are as follows:

$$0.1 \leq l_1 \leq 2,$$
$$0.1 \leq l_2 \leq 10,$$
$$0.1 \leq l_3 \leq 10,$$
$$0.1 \leq l_4 \leq 2$$

Where,

$$\tau\left(\vec{l}\right) = \sqrt{(\tau')^2 + 2\tau'\tau''\left(\frac{l_2}{R}\right) + (\tau'')^2}$$

$$\tau' = P/\sqrt{2 l_1 l_2}$$

$$\tau'' = MR/J$$

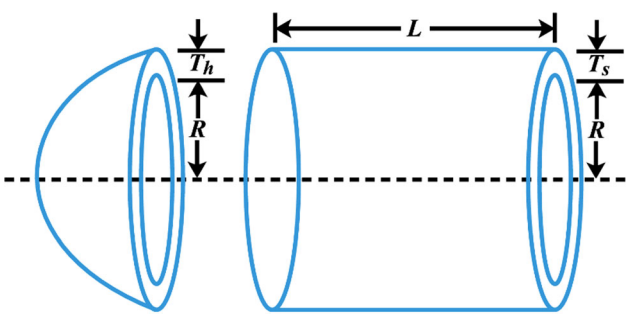


**Fig. 7** Pressure vessel design problem

$$M = P\left(L + \frac{l_2}{2}\right)$$

$$R = \sqrt{\frac{l_2^2}{4} + \left(\frac{l_1 + l_3}{2}\right)^2}$$

$$J = 2\left\{\sqrt{2} l_1 l_2 \left[\left(\frac{(l_2^2)}{12}\right) + \left(\frac{l_1 + l_3}{2}\right)^2\right]\right\}$$

$$\sigma\left(\vec{l}\right) = \frac{6PL}{l_4 l_3^2}$$

$$\delta\left(\vec{l}\right) = \frac{4PL^3}{E l_3^3 l_4}$$

$$P_c\left(\vec{l}\right) = \frac{4.013E\sqrt{\frac{l_3^2 l_4^6}{36}}}{L^2}\left(1 - \frac{l_3}{2L}\sqrt{E/4G}\right).$$

The values for parameters of WBD are as follows:

$$\sigma_{max} = 30000\,psi, \mathrm{P} = 6000\mathrm{lb}, \mathrm{L} = 14\ \mathrm{in}, \delta_{max} = 0.25\,in, E = 30 \times 10^6 psi, \tau_{max} = 13600\,psi, \text{and G} = 12 \times 10^6\,psi.$$

#### 4.2.2 Compression spring design problem

The compression spring design problem (CSD) falls under the continuous constrained optimization problem category. The GOA aims to find the minimum volume $V$ of a coil spring under a constant tension/compression load. The CSD is shown in Fig. 6, and the three design variables are as follows:

- The number of compression spring's active coils $P = \ \times 1 \in [2, 15]$
- The diameter of the spring's winding $D = \ \times 2 \in [0.25, 1.3]$
- The diameter of the wire $d = \ \times 3 \in [0.05, 2]$

The CSD problem is modeled as in Eq. 13 (Kazemzadeh-Parsi 2014). Given that

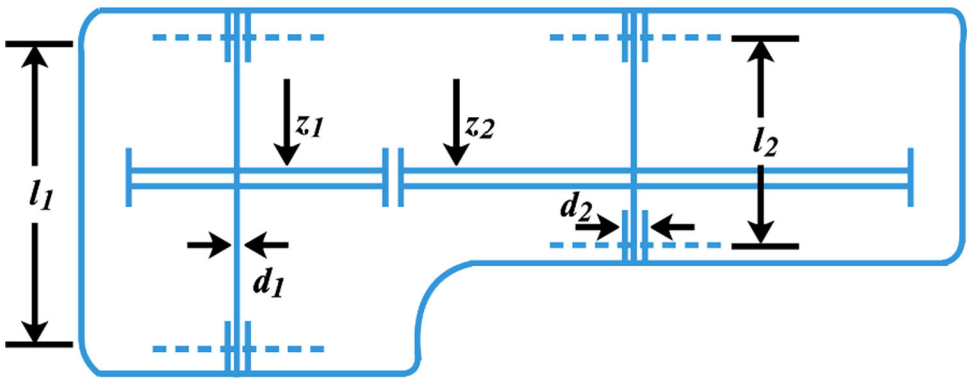


**Fig. 8** The speed reducer design problem

$$l = [l_1 l_2 l_3] = [\text{dDP}]\,\text{Min}f\left(\vec{l}\right) = (l_3 + 2) l_2 l_1^2 \tag{13}$$

subject to

$$s_1(\overrightarrow{l}) = 1 - \frac{l_2^3 l_3}{71,785 l_1^4} \leq 0,$$

$$s_2(\overrightarrow{l}) = \frac{4l_2^2 - l_1 l_2}{12,566\left(l_3 l_1^3 - l_1^4\right)} + \frac{1}{5108 l_1^2} \leq 0,$$

$$s_3(\overrightarrow{l}) = 1 - \frac{140.45 l_1}{l_2^2 l_3} \leq 0,$$

$$s_4(\overrightarrow{l}) = \frac{l_1 + l_2}{1.5} - 1 \leq 0$$

The intervals for the design variables are:

$$0.05 \leq l_1 \leq 2.00,$$
$$0.25 \leq l_2 \leq 1.30,$$
$$2.00 \leq l_3 \leq 15.0$$

#### 4.2.3 Pressure vessel design problem

A pressure vessel design model (PVD) consists of a cylinder closed with end caps shown in Fig. 7. The PVD has four decision variables where $x1$ is the thickness of the pressure vessel $T_s$, $x2$ is the thickness of the head $T_h$,$x3$ stands for the inner radius of the vessel $R$, and $x4$ is the length of the vessel barring head $L$. The mathematical model for PVD is given in Eq. 14 (Sandgren, 1990). Given that

$$\begin{aligned} l &= [l_1 l_2 l_3 l_4] = [T_s T_h R L], \\ &\text{Min}f(\vec{l}) = 0.6224 l_1 l_3 l_4 + 1.781 l_2 l_3^2 \\ &+ 3.1661 l_1^2 l_4 + 19.84 l_1^2 l_3 \end{aligned} \tag{14}$$

$$s_1\left(\vec{l}\right) = -l_1 + 0.0193 l_3 \leq 0,$$

$$s_2\left(\vec{l}\right) = -l_2 + 0.00954 l_3 \leq 0,$$

$$s_3\left(\vec{l}\right) = -\pi l_3^2 l_4 - \frac{4}{3}\pi l_3^3 + 1,296,000 \leq 0,$$

$$s_4\left(\vec{l}\right) = l_4 - 240 \leq 0.$$

The interval of the design variables are as follows:

$$0 \leq l_1 \leq 99,$$
$$0 \leq l_2 \leq 99,$$
$$10 \leq l_3 \leq 200,$$
$$10 \leq l_4 200.$$

#### 4.2.4 The speed reducer design problem (SRD)

Speed reducers, sometimes called the Gear-reducer, are a component found in mechanical, automobile, electrical, and hydraulic motors. They are different combinations of gears designed to increase the torque of a motor. The SRD shown in Fig. 8 is a constrained optimization problem where the goal is to minimize the weight of the gearbox subject to satisfying several gears and shaft design constraints [40]. There are seven (7) design variables where $x_1$ is the face width (b), $x_2$ represents the module of teeth (m), $x_3$ is the number of teeth in the pinion (z), $x_4$ denotes the length of the first shaft between bearings $(l_1)$, length of the second shaft between the bearings $(l_2)$ is represented by $x_5$, and the diameter of first $(d_1)$ and second shafts $(d_2)$ are denoted by $x_6$ and $x_7$, respectively. This problem is modeled mathematically as given in Eq. 15. Given that

$$x = [x_1, x_2, x_3, x_4, x_5, x_6, x_7] = [b, m, z, l_1, l_2, d_1, d_2]$$

$$\begin{aligned} \min f(x) &= 0.7854 x_1 x_2^2 \left(10/x_2^3 + 14.9334 x_3 - 43.0934\right) \\ &- 1.508 x_1 \left(x_6^2 + x_7^2\right) \\ &+ 7.4777\left(x_6^3 + x_7^3\right) + 0.7854\left(x_4 x_6^2 + x_5 x_7^2\right) \end{aligned} \tag{15}$$

**Table 4** Classical test functions results

| Function | Value | GOA | nAOA | AOA | CPSOGSA | PSO | BBO | DE | SSA | SCA | GWO |
|---|---|---|---|---|---|---|---|---|---|---|---|
| F1 | Best | 0 | 0 | 0 | 0 | 9.5172 | 2.29E-05 | 42,925 | 2.63E-08 | 0 | 0 |
| | Worst | 0 | 0 | 0 | 10,000 | 18.136 | 8.67E-05 | 42,925 | 5.67E-08 | 696.27 | 0 |
| | Average | 0 | 0 | 0 | 333.34 | 13.444 | 4.19E-05 | 42,925 | 4.11E-08 | 54.54 | 0 |
| | SD | 0 | 0 | 0 | 1825.7 | 1.938 | 1.40E-05 | 0 | 7.65E-09 | 131.35 | 0 |
| | Median | 0 | 0 | 0 | 0 | 13.215 | 3.97E-05 | 42,925 | 4.09E-08 | 12.608 | 0 |
| F2 | Best | 0 | 0 | 0 | 1.33E-08 | 1.1176 | 0.000406 | 3.04E + 64 | 0.50286 | 0 | 0 |
| | Worst | 0 | 0 | 0 | 37.179 | 1.8525 | 0.000786 | 3.04E + 64 | 5.577 | 0.088169 | 0 |
| | Average | 0 | 0 | 0 | 6.5001 | 1.5457 | 0.000577 | 3.04E + 64 | 2.2844 | 0.008764 | 0 |
| | SD | 0 | 0 | 0 | 10.193 | 0.16489 | 9.91E-05 | 2.10E + 49 | 1.3134 | 0.017489 | 0 |
| | Median | 0 | 0 | 0 | 2.12 | 1.5591 | 0.000578 | 3.04E + 64 | 2.2475 | 0.002618 | 0 |
| F3 | Best | 0 | 0 | 0.028066 | 4682.7 | 763.79 | 75,123 | 2.04E + 07 | 492.25 | 0 | 0 |
| | Worst | 0 | 37.769 | 0.028066 | 19,313 | 1645.6 | 1.04E + 05 | 2.16E + 07 | 5346.2 | 53,336 | 8.43E-08 |
| | Average | 0 | 3.5291 | 0.028066 | 11,217 | 1196.1 | 92,273 | 2.10E + 07 | 2064.9 | 30,544 | 4.63E-09 |
| | SD | 0 | 7.4743 | 0 | 3878 | 232 | 6863.9 | 3.55E + 05 | 1031.9 | 8914.2 | 1.66E-08 |
| | Median | 0 | 0.001567 | 0.028066 | 11,018 | 1186.4 | 93,192 | 2.09E + 07 | 1674.5 | 30,143 | 0 |
| F4 | Best | 0 | 0 | 0.048503 | 36.83 | 1.7755 | 17.173 | 50 | 7.3504 | 0 | 0 |
| | Worst | 0 | 0.000753 | 0.048503 | 91.583 | 2.337 | 25.5 | 50 | 17.661 | 72.124 | 0 |
| | Average | 0 | 2.85E-05 | 0.048503 | 71.609 | 2.0562 | 22.31 | 50 | 12.889 | 57.926 | 0 |
| | SD | 0 | 0.000138 | 0 | 21.181 | 0.1326 | 1.9203 | 0 | 2.6895 | 6.3537 | 0 |
| | Median | 0 | 0 | 0.048503 | 84.441 | 2.0756 | 22.446 | 50 | 12.579 | 57.157 | 0 |
| F5 | Best | 42.474 | 48.423 | 48.589 | 37.48 | 215.33 | 94.204 | 5.68E + 09 | 44.065 | 42.474 | 45.491 |
| | Worst | 45.681 | 48.423 | 48.589 | 333.44 | 1146.7 | 345.16 | 5.70E + 09 | 917.32 | 3.22E + 06 | 48.618 |
| | Average | 44.392 | 48.423 | 48.589 | 86.64 | 388.35 | 198.38 | 5.70E + 09 | 123.75 | 4.03E + 05 | 46.925 |
| | SD | 0.60983 | 0 | 0 | 72.325 | 196.54 | 58.302 | 5.51E + 06 | 165.11 | 6.98E + 05 | 0.82685 |
| | Median | 44.408 | 48.423 | 48.589 | 48.218 | 336.48 | 186.17 | 5.70E + 09 | 86.282 | 1.12E + 05 | 47.031 |
| F6 | Best | 0.001075 | 3.2651 | 6.3098 | 0 | 7.9195 | 2.07E-05 | 44,213 | 2.29E-08 | 0.001075 | 0.4075 |
| | Worst | 0.97399 | 3.2651 | 6.3098 | 10,100 | 17.546 | 5.60E-05 | 44,213 | 5.49E-08 | 108.77 | 2.254 |
| | Average | 0.45976 | 3.2651 | 6.3098 | 336.68 | 13.459 | 3.92E-05 | 44,213 | 3.92E-08 | 31.097 | 1.5687 |
| | SD | 0.26899 | 0 | 0 | 1844 | 2.228 | 1.03E-05 | 0 | 8.26E-09 | 23.44 | 0.48281 |
| | Median | 0.52767 | 3.2651 | 6.3098 | 0 | 13.64 | 3.90E-05 | 44,213 | 3.78E-08 | 23.937 | 1.6134 |
| F7 | Best | 4.14E-05 | 0.000399 | 2.10E-05 | 0.073662 | 0.005496 | 0.048251 | 9.22E + 08 | 0.081729 | 0.000399 | 0.000139 |
| | Worst | 4.14E-05 | 0.007156 | 2.10E-05 | 0.20935 | 0.01469 | 0.097603 | 1.03E + 09 | 0.28857 | 3.5654 | 0.00159 |
| | Average | 4.14E-05 | 0.001537 | 2.10E-05 | 0.13048 | 0.009487 | 0.07612 | 9.83E + 08 | 0.17227 | 0.55889 | 0.00075 |
| | SD | 0 | 0.001296 | 0 | 0.029784 | 0.002134 | 0.013977 | 3.13E + 07 | 0.059453 | 0.70736 | 0.000363 |
| | Median | 4.14E-05 | 0.001246 | 2.10E-05 | 0.12726 | 0.009685 | 0.072471 | 9.88E + 08 | 0.16779 | 0.29078 | 0.0007 |

**Table 4** (continued)

| Function | Value | GOA | nAOA | AOA | CPSOGSA | PSO | BBO | DE | SSA | SCA | GWO |
|---|---|---|---|---|---|---|---|---|---|---|---|
| F8 | Best | − 9520.9 | − 8119.8 | − 14,685 | − 13,887 | − 14,357 | − 14,017 | 216.79 | − 15,000 | − 14,685 | − 11,538 |
| | Worst | − 9520.9 | − 8119.8 | − 9703.6 | − 9948.7 | − 11,706 | − 12,282 | 216.79 | − 9282.5 | − 4692.2 | − 4207.9 |
| | Average | − 9520.9 | − 8119.8 | − 11,714 | − 12,072 | − 13,148 | − 13,206 | 216.79 | − 12,151 | − 5265.9 | − 9177.7 |
| | SD | 0 | 0 | 1383 | 999.81 | 661.03 | 401.9 | 1.04E − 13 | 1051.4 | 382.11 | 1291.2 |
| | Median | − 9520.9 | − 8119.8 | − 11,410 | − 12,165 | − 13,206 | − 13,114 | 216.79 | − 12,285 | − 5098 | − 9183.1 |
| F9 | Best | 0 | 0 | 0 | 101.49 | 44.375 | 180.45 | 42,925 | 27.859 | 0 | 0 |
| | Worst | 0 | 75.997 | 0 | 286.55 | 104.22 | 216.55 | 42,925 | 115.42 | 255.17 | 0 |
| | Average | 0 | 2.5332 | 0 | 207.91 | 77.126 | 200.5 | 42,925 | 67.392 | 63.554 | 0 |
| | SD | 0 | 13.875 | 0 | 50.507 | 16.854 | 9.1328 | 0 | 25.928 | 51.514 | 0 |
| | Median | 0 | 0 | 0 | 209.44 | 79.097 | 201.05 | 42,925 | 61.19 | 53.079 | 0 |
| F10 | Best | 0 | 0 | 0 | 0 | 0.84275 | 0.001041 | 19.943 | 1.3743 | 0 | 0 |
| | Worst | 0 | 0 | 0 | 19.428 | 1.421 | 0.001561 | 19.943 | 4.449 | 20.452 | 0 |
| | Average | 0 | 0 | 0 | 9.6014 | 1.1896 | 0.00132 | 19.943 | 2.7351 | 16.422 | 0 |
| | SD | 0 | 0 | 0 | 7.9606 | 0.15145 | 0.000135 | 3.61E-15 | 0.73797 | 8.0367 | 0 |
| | Median | 0 | 0 | 0 | 11.214 | 1.1812 | 0.001297 | 19.943 | 2.5792 | 20.354 | 0 |
| F11 | Best | 0 | 0 | 0.16318 | 0.62438 | 1.0795 | 2.05E-05 | 11.731 | 2.37E-06 | 0 | 0 |
| | Worst | 0 | 0 | 0.16318 | 181.03 | 1.1639 | 0.000633 | 11.731 | 0.024869 | 4.3584 | 0.016052 |
| | Average | 0 | 0 | 0.16318 | 16.126 | 1.1178 | 0.000102 | 11.731 | 0.005679 | 1.1952 | 0.001698 |
| | SD | 0 | 0 | 0 | 38.218 | 0.021276 | 0.000111 | 2.98E-07 | 0.007263 | 0.75567 | 0.004534 |
| | Median | 0 | 0 | 0.16318 | 4.1845 | 1.1171 | 6.94E-05 | 11.731 | 0.000278 | 1.0596 | 0 |
| F12 | Best | 6.00E-05 | 0.26752 | 0.57062 | 5.2018 | 0.014925 | 7.72E-06 | 2.18E + 09 | 2.892 | 6.00E-05 | 0.017852 |
| | Worst | 0.022871 | 0.26752 | 0.57062 | 16.931 | 0.033278 | 4.91E-05 | 2.18E + 09 | 13.398 | 3.61E + 06 | 0.11162 |
| | Average | 0.007391 | 0.26752 | 0.57062 | 8.9293 | 0.023089 | 2.42E-05 | 2.18E + 09 | 6.821 | 3.47E + 05 | 0.064446 |
| | SD | 0.006435 | 0 | 0 | 2.938 | 0.005007 | 1.07E-05 | 19.298 | 2.3759 | 8.27E + 05 | 0.025162 |
| | Median | 0.006151 | 0.26752 | 0.57062 | 8.9987 | 0.022166 | 2.26E-05 | 2.18E + 09 | 6.0235 | 2622.4 | 0.064121 |
| F13 | Best | 0.024287 | 3.991 | 4.7546 | 0 | 0.31685 | 5.20E-05 | 3.90E + 09 | 1.94E-05 | 0.024287 | 0.79601 |
| | Worst | 0.70659 | 3.991 | 4.7546 | 54.71 | 0.70711 | 0.000199 | 3.90E + 09 | 74.103 | 7.60E + 06 | 1.8762 |
| | Average | 0.32705 | 3.991 | 4.7546 | 20.918 | 0.53361 | 0.00012 | 3.90E + 09 | 29.587 | 6.09E + 05 | 1.3201 |
| | SD | 0.17498 | 0 | 0 | 15.471 | 0.097484 | 3.55E-05 | 0 | 21.985 | 1.44E + 06 | 0.31049 |
| | Median | 0.34353 | 3.991 | 4.7546 | 20.555 | 0.53472 | 0.000126 | 3.90E + 09 | 30.517 | 1.08E + 05 | 1.3115 |
| F14 | Best | 0.000307 | 0.000687 | 0.000377 | 0.000331 | 0.000308 | 0.000371 | 0.40836 | 0.00035 | 0.000307 | 0.000307 |
| | Worst | 0.000307 | 0.000687 | 0.000377 | 0.020363 | 0.020363 | 0.000752 | 0.40836 | 0.001413 | 0.00146 | 0.056543 |
| | Average | 0.000307 | 0.000687 | 0.000377 | 0.004082 | 0.002778 | 0.000631 | 0.40836 | 0.000773 | 0.000838 | 0.003696 |
| | SD | 9.57E-15 | 0 | 0 | 0.007412 | 0.006007 | 0.000112 | 1.13E-16 | 0.000279 | 0.000388 | 0.011185 |
| | Median | 0.000307 | 0.000687 | 0.000377 | 0.000745 | 0.000664 | 0.000668 | 0.40836 | 0.000737 | 0.000681 | 0.000307 |

**Table 4** (continued)

| Function | Value | GOA | nAOA | AOA | CPSOGSA | PSO | BBO | DE | SSA | SCA | GWO |
|---|---|---|---|---|---|---|---|---|---|---|---|
| F15 | Best | 3 | 3 | 3 | 3 | 3 | 3 | 2275 | 3 | 3 | 3 |
| | Worst | 3 | 3 | 3 | 3 | 30 | 3 | 2275 | 3 | 3 | 3 |
| | Average | 3 | 3 | 3 | 3 | 4.8 | 3 | 2275 | 3 | 3 | 3 |
| | SD | 0 | 6.70E-16 | 0 | 1.57E-15 | 6.8501 | 1.22E-15 | 0 | 6.63E-14 | 8.71E-06 | 3.17E-06 |
| | Median | 3 | 3 | 3 | 3 | 3 | 3 | 2275 | 3 | 3 | 3 |

Subject to

$$g_1(x) = \frac{27}{x_1 x_2^2 x_3} - 1 \le 0,$$

$$g_2(x) = \frac{397.5}{x_1 x_2^2 x_3^2} - 1 \le 0,$$

$$g_3(x) = \frac{1.93 x_4^2}{x_2 x_6^4 x_3} - 1 \le 0,$$

$$g_4(x) = \frac{1.93 x_5^2}{x_2 x_7^4 x_3} - 1 \le 0,$$

$$g_5(x) = \frac{\sqrt{\left(\frac{745 x_4}{x_2 x_3}\right)^2 + 16 \times 10^6}}{110 x_6^3} - 1 \le 0,$$

$$g_6(x) = \frac{\sqrt{\left(\frac{745 x_5}{x_2 x_3}\right)^2 + 157.5 \times 10^6}}{85 x_7^3} - 1 \le 0,$$

$$g_7(x) = \frac{x_2 x_3}{40} - 1 \le 0,$$

$$g_8(x) = \frac{5 x_2}{x_1} - 1 \le 0,$$

$$g_9(x) = \frac{x_1}{12 x_2} - 1 \le 0,$$

$$g_{10}(x) = \frac{1.5 x_6 + 1.9}{x_4} - 1 \le 0,$$

$$g_{11}(x) = \frac{1.5 x_7 + 1.9}{x_5} - 1 \le 0$$

The interval is given as follows:

$$2.6 \le x_1 \le 3.6, 0.7 \le x_2 \le 0.8, 17 \le x_3 \le 28, 7.3 \le x_4 \le 8.3,$$
$$7.3 \le x_5 \le 8.3, 2.9 \le x_6 \le 3.9, 5.0 \le x_7 \le 5.5$$

## 5 Results and discussion

This section presents and discusses the results of the experiments conducted on the benchmark functions and the four (4) engineering problems. The results were presented using the 'best,' 'worst,' 'average,' 'standard deviation (SD),' and 'median' indicators. All the algorithms are compared using mean, standard deviation, Friedman's test, and Wilcoxon signed-rank test to check the performance differences between all algorithms for all functions. Friedman's test assumes a 0.05 significance level and the null hypothesis that there is no significant difference between the mean results of all the participating algorithms. The alternative hypothesis is that there is a significant difference in the mean results of the participating algorithms. The Wilcoxon signed-rank test set a 0.05 significance level, $R+$ represents the sum of ranks in which the algorithm appearing first outperforms the algorithm

**Table 5** Results for composite functions (CEC2020)

| Function | Value | GOA | nAOA | AOA | CPSOGSA | PSO | BBO | DE | SSA | SCA | GWO |
|---|---|---|---|---|---|---|---|---|---|---|---|
| F16 | Best | 100.04 | 3.89E + 08 | 1.62E + 09 | 101.38 | 101.9 | 103.43 | 199.57 | 147.43 | 100.04 | 5734.3 |
| | Worst | 101.51 | 7.61E + 09 | 1.03E + 10 | 12,725 | 8494.5 | 9805 | 6437.4 | 9096.2 | 1.91E + 09 | 3.43E + 08 |
| | Average | 100.43 | 3.42E + 09 | 5.11E + 09 | 5175.8 | 2008.6 | 1343.2 | 1759.2 | 2893.4 | 6.96E + 08 | 1.33E + 07 |
| | SD | 0.35345 | 1.75E + 09 | 2.22E + 09 | 3826.3 | 2336.6 | 2201.4 | 1467.8 | 2845.9 | 3.24E + 08 | 6.28E + 07 |
| | Median | 100.35 | 3.36E + 09 | 5.35E + 09 | 5158.1 | 1236.3 | 285.98 | 1250.8 | 1756.6 | 6.69E + 08 | 21,409 |
| F17 | Best | 1106.6 | 1352.5 | 1266 | 1439.3 | 1107 | 1113.6 | 1246.5 | 1320.7 | 1106.6 | 1116.7 |
| | Worst | 1495.8 | 2693 | 2787.2 | 2642.2 | 1709.9 | 2115 | 1804.1 | 2378.9 | 2760.4 | 1941.2 |
| | Average | 1246.4 | 2073.7 | 2124 | 2149.9 | 1399.6 | 1681.2 | 1540.9 | 1861.6 | 2308.3 | 1535.4 |
| | SD | 107.35 | 291.63 | 329.5 | 355.12 | 137.58 | 244.51 | 130.05 | 276.66 | 224.26 | 233.62 |
| | Median | 1246.8 | 2121 | 2123.4 | 2182.9 | 1445 | 1704.3 | 1557.6 | 1886 | 2386.9 | 1475.1 |
| F18 | Best | 711.23 | 733.87 | 759.98 | 722.29 | 703.6 | 715.05 | 716.7 | 716.96 | 711.23 | 714.36 |
| | Worst | 724.6 | 816.59 | 821.47 | 788.59 | 725.1 | 735.73 | 725.64 | 772.3 | 802.59 | 749.27 |
| | Average | 719.59 | 766.69 | 798.12 | 752.59 | 716.98 | 722.02 | 721.45 | 734.23 | 776.62 | 727.71 |
| | SD | 3.0154 | 22.882 | 17.539 | 16.364 | 4.9699 | 4.9793 | 2.3513 | 12.331 | 9.9176 | 9.3905 |
| | Median | 720.2 | 762.72 | 803.36 | 754.74 | 717.67 | 722.41 | 721.17 | 730.78 | 775.65 | 727.69 |
| F19 | Best | 1900.5 | 2402.4 | 3735.6 | 1900.6 | 1900.3 | 1900.5 | 1901.1 | 1900.8 | 1900.5 | 1900.7 |
| | Worst | 1901.8 | 1.32E + 05 | 2.77E + 05 | 1903.7 | 1902 | 1906.5 | 1902.2 | 1904.4 | 1941.3 | 1904.1 |
| | Average | 1901.2 | 31,536 | 50,020 | 1901.8 | 1901.1 | 1901.9 | 1901.6 | 1901.6 | 1919 | 1902.4 |
| | SD | 0.31242 | 29,526 | 56,624 | 0.83199 | 0.46064 | 1.2235 | 0.25486 | 0.9302 | 8.2064 | 0.75162 |
| | Median | 1901.1 | 25,638 | 37,525 | 1901.6 | 1901 | 1901.5 | 1901.6 | 1901.2 | 1916.7 | 1902.5 |
| F20 | Best | 1734.7 | 6757.7 | 3968.2 | 1994.7 | 1871.2 | 2461 | 4359.2 | 2681.8 | 1734.7 | 3029.7 |
| | Worst | 1799.6 | 5.95E + 05 | 6.78E + 05 | 1.56E + 05 | 10,218 | 1.31E + 05 | 80,032 | 15,641 | 77,282 | 6.11E + 05 |
| | Average | 1756.1 | 1.51E + 05 | 1.43E + 05 | 14,147 | 4709.3 | 24,327 | 28,146 | 5928.7 | 31,187 | 69,600 |
| | SD | 16.75 | 1.36E + 05 | 1.33E + 05 | 29,030 | 2794.8 | 33,785 | 17,680 | 3600.6 | 21,634 | 1.69E + 05 |
| | Median | 1752 | 1.26E + 05 | 1.42E + 05 | 6575.3 | 3619 | 10,825 | 23,485 | 4417.1 | 23,966 | 5745.1 |
| F21 | Best | 1600 | 1600.8 | 1600.5 | 1600 | 1600 | 1600 | 1600 | 1600 | 1600 | 1600 |
| | Worst | 1600 | 1659.2 | 1622.9 | 1724.9 | 1658.5 | 1617.6 | 1600.4 | 1617.1 | 1601.7 | 1625.5 |
| | Average | 1600 | 1617.3 | 1611.5 | 1630.3 | 1612.6 | 1602.8 | 1600.2 | 1601.6 | 1601 | 1603 |
| | SD | 0.000893 | 18.114 | 8.3504 | 35.718 | 14.912 | 5.8203 | 0.10355 | 4.2149 | 0.21847 | 6.6623 |
| | Median | 1600 | 1617.2 | 1616 | 1616.8 | 1616.8 | 1600.6 | 1600.2 | 1600.5 | 1601 | 1600.3 |
| F22 | Best | 2102.9 | 2750.9 | 2768.6 | 2341.5 | 2217.7 | 2161.5 | 2271.8 | 2424.6 | 2102.9 | 2708.6 |
| | Worst | 2122.5 | 30,182 | 9834.9 | 22,472 | 4542.1 | 28,362 | 8615.2 | 26,567 | 15,558 | 18,025 |
| | Average | 2110 | 9998.4 | 6090.4 | 5808.2 | 2718.8 | 9738.9 | 3340.3 | 6743.2 | 7865 | 9151.4 |
| | SD | 5.5831 | 7532.6 | 2245.1 | 4995.4 | 456.78 | 7763.2 | 1260.2 | 6332.9 | 3543 | 4892.4 |
| | Median | 2108.4 | 7917.4 | 6060 | 3030.7 | 2742.5 | 7488.6 | 2992 | 3418.5 | 7416 | 7390.2 |

**Table 5** (continued)

| Function | Value | GOA | nAOA | AOA | CPSOGSA | PSO | BBO | DE | SSA | SCA | GWO |
|---|---|---|---|---|---|---|---|---|---|---|---|
| F23 | Best | 2200.2 | 2361.5 | 2478.1 | 2300.6 | 2300.7 | 2300.3 | 2261.9 | 2223 | 2200.2 | 2300.9 |
| | Worst | 2303.5 | 2665.4 | 3865.3 | 2308.9 | 2892.2 | 2304.6 | 2302.1 | 2308.8 | 2502 | 2322.5 |
| | Average | 2286.7 | 2504.8 | 2786.2 | 2302.7 | 2321.6 | 2302 | 2298.8 | 2299 | 2374.9 | 2304.7 |
| | SD | 35.252 | 71.753 | 274.14 | 1.5989 | 107.77 | 1.043 | 7.5443 | 18.101 | 39.318 | 4.7891 |
| | Median | 2301.8 | 2500.9 | 2743.1 | 2302.6 | 2301.9 | 2301.8 | 2300.7 | 2303.1 | 2369.6 | 2302.4 |
| F24 | Best | 2418.4 | 2500.2 | 2569.3 | 2500 | 2729.4 | 2400 | 2646.2 | 2500 | 2418.4 | 2502 |
| | Worst | 2734.1 | 2892.6 | 2892.7 | 2816.8 | 2756.2 | 2760.4 | 2755.4 | 2786.3 | 2798.3 | 2774.7 |
| | Average | 2523.3 | 2815 | 2799.1 | 2755 | 2740.5 | 2725.3 | 2737 | 2740.7 | 2774.8 | 2735.5 |
| | SD | 67.32 | 80.696 | 78.55 | 78.937 | 6.0925 | 76.252 | 23.448 | 46.867 | 40.458 | 45.785 |
| | Median | 2500.2 | 2827 | 2819.1 | 2772.8 | 2740.9 | 2743.3 | 2745 | 2746.3 | 2783 | 2740.1 |
| F25 | Best | 2601.3 | 2903.7 | 2929.6 | 2897.8 | 2897.8 | 2897.8 | 2899.2 | 2897.8 | 2601.3 | 2898.1 |
| | Worst | 2897.8 | 3218.2 | 3625.1 | 3024.2 | 2947.1 | 2950.6 | 2946.6 | 2949.3 | 2988.6 | 2949.7 |
| | Average | 2871.8 | 3005.8 | 3171.8 | 2939.8 | 2921.6 | 2932 | 2918.5 | 2918.8 | 2962.1 | 2937.3 |
| | SD | 80.727 | 67.982 | 148.27 | 41.001 | 23.392 | 22.204 | 14.58 | 23.897 | 12.341 | 17.896 |
| | Median | 2897.7 | 2983.4 | 3140.7 | 2943.9 | 2921.7 | 2945.1 | 2917.2 | 2898.8 | 2961.8 | 2946 |

appearing second, and $R-$ denotes the sum of ranks for which the algorithm appearing second outperforms the algorithm appearing first. The null hypothesis is that there is no significant difference between the mean results of the two participating algorithms. The alternative hypothesis is that there is a significant difference in the mean results of the two participating algorithms.

### 5.1 Results for benchmark functions

The classical and composite test function evaluates how the GOA escapes from local optima and performs exploitation and exploration. The unimodal functions F1–F7 are perfect for testing the exploitation capability of GOA. They have only one global optimum. Looking at Table 4, it can be seen that GOA was able to find the global optimum for F1-F4 and near-optimal for F5-F7, with observed premature convergence in F5 and F6. Similarly, functions F8–F15 have many local optima located randomly within the search space. These functions are well suited for exploration tests. The GOA, in most cases, is observed to return the least values for the 'mean' and 'standard-deviation' indicators. The composite functions (CEC2020) provide a test for the ability of GOA to escape local minima. It can be seen from Table 5 that GOA found the optimal solution for F16, F19, F21, and F23 and near-optimal for the remaining functions. In all, the GOA also returned the least values for the 'mean and standard deviation' indicators and outperformed the nine algorithms considered for all the functions. The convergence graphs in Fig. 9 illustrate this advantage graphically for both the classical and composite functions. It is seen that the proposed algorithm converges steadily towards the optimal result for most functions.

Though GWO, nAOA, and AOA were very competitive, finding the optimal solutions accordingly, the superiority of GOA is confirmed by the result of Friedman's test and Wilcoxon signed test. The mean values were used for Friedman's test. A p-value of 0.00 returned, less than the 0.05 significance level, indicates that the obtained results for classical test functions for all the algorithms considered are significantly different. The mean ranks and other test statistics for classical test functions are presented in Table 6. The GOA clearly returned the lowest mean rank value, closely followed by GWO, AOA, and nAOA. This implies that the GOA is ranked first, GWO is ranked second, AOA is ranked third, and the rest are ranked, as shown in Table 6. The results for the Wilcoxon signed test on classical benchmark test functions are presented in Table 7. It can be seen that GOA obtained higher $R+$ values than $R-$ or ties in all the cases. This implies that GOA significantly outperforms nAOA, CPSOGSA, PSO, DE, SSA, SCA, and GWO for all of the classical

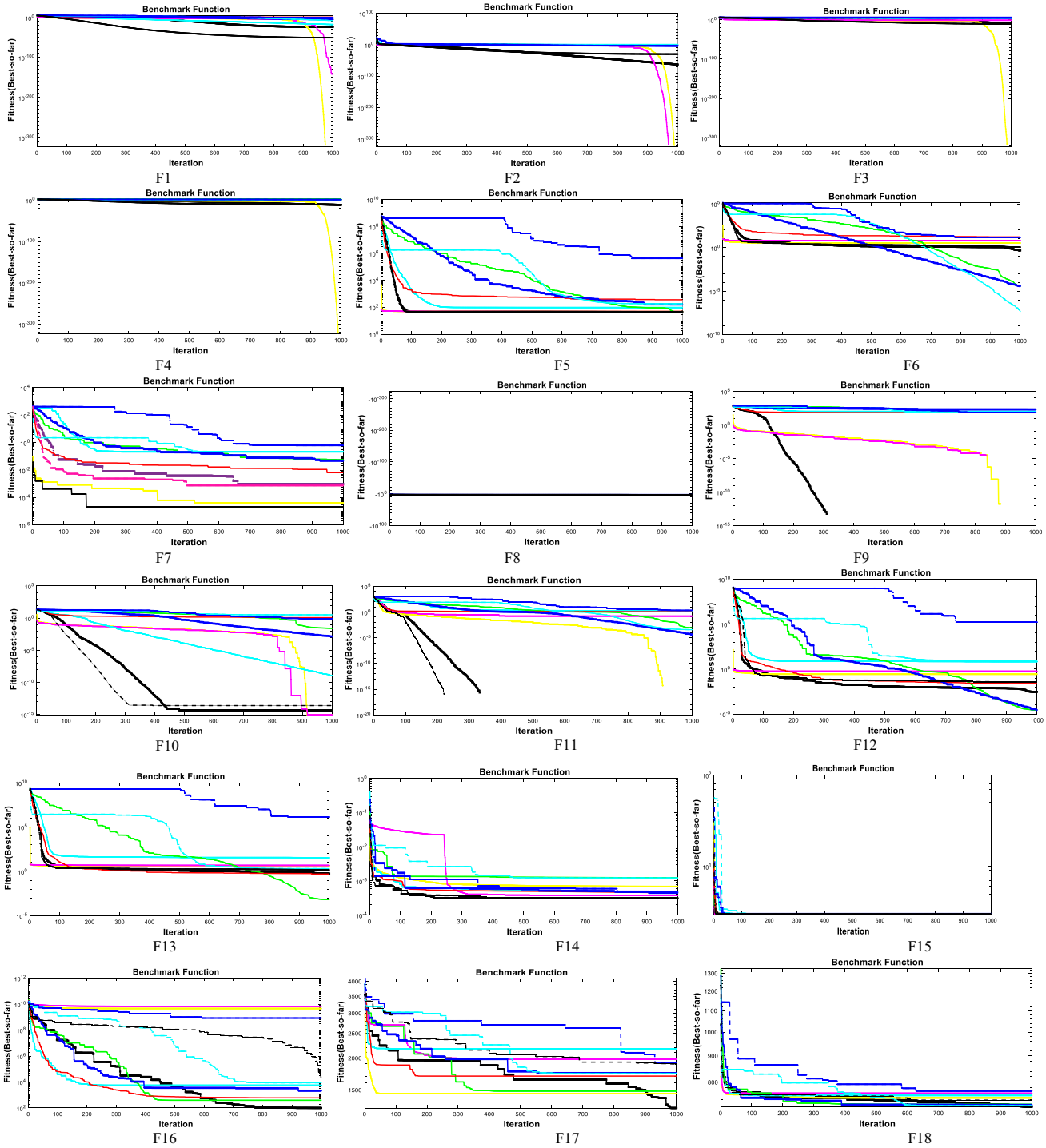


**Fig. 9** Convergence for classical and composite functions

benchmark test functions. According to the p-values returned by Wilcoxon's test at $\alpha = 0.05$, there is a significant difference in the performance of the algorithms in 8 out of 9 cases, which means that GOA significantly outperforms 8 out of 9 algorithms on 15 test functions. Furthermore, the GOA is less than, equal to, or better than other algorithms in 10, 20, and 104 out of 134 cases.

In the same vein, the mean ranks and other test statistics for CEC 2020 test functions are presented in Table 8. A *p*-value of 0.00 returned, less than the 0.05 significance level, indicates that the obtained results for all the algorithms

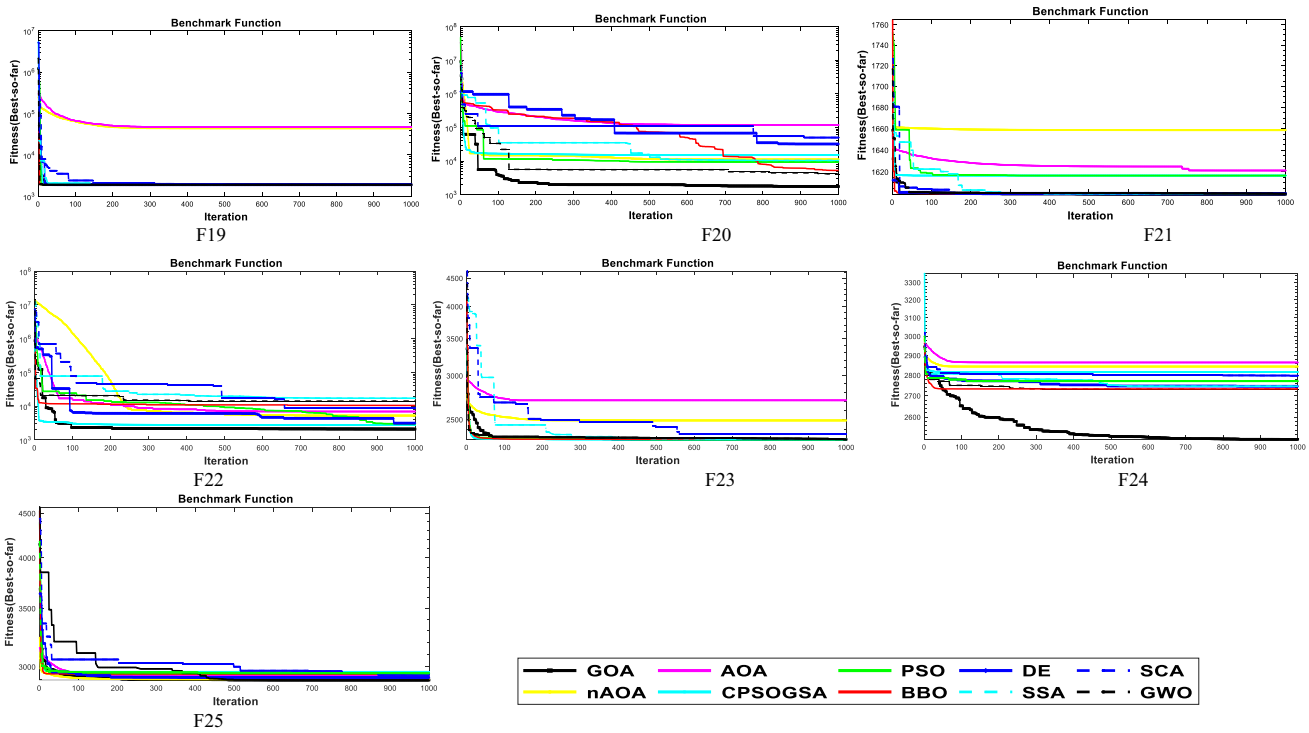


**Fig. 9** continued

**Table 6** Friedman's test results for classical test functions

| Algorithm | Mean rank | Rank |
|---|---|---|
| GOA | 2.57 | 1 |
| nAOA | 4.32 | 4 |
| AOA | 4.29 | 3 |
| CPSOGSA | 8.43 | 8 |
| PSO | 6.29 | 7 |
| BBO | 4.86 | 5 |
| DE | 10.79 | 10 |
| SSA | 6.07 | 6 |
| SCA | 8.71 | 9 |
| GWO | 3.89 | 2 |
| *Test statistics* | | |
| N | 14 | |
| Chi-square | 79.574 | |
| Df | 10 | |
| *P*-value | 0.000 | |

considered are significantly different. The GOA returned the lowest mean rank value, closely followed by DE, PSO, and SSA. This implies that the GOA is ranked first, DE is ranked second, PSO is ranked third, and the rest are ranked, as shown in Table 8. The results for the Wilcoxon signed test on CEC 2020 test functions are presented in Table 9. It can be seen that GOA obtained higher $R+$ values than $R-$ or ties in all the cases. This implies that GOA significantly outperforms all the algorithms used in the comparison for all the CEC 2020 test functions. According to the p-values returned by Wilcoxon's test at $\alpha = 0.05$, there is a significant difference in the performance of the algorithms in all cases, which means that GOA significantly outperforms the 9 algorithms on 10 test functions. Furthermore, the GOA is less than, equal to, or better than other algorithms in 2, 0, and 88 out of 90 cases. Thus, it can be concluded that the performance of GOA is better in terms of search quality, efficiency, and robustness.

## 5.2 Results for engineering problems

As explained previously, the proposed GOA was used to solve four (4) mechanical engineering design problems. The constraint handling was handled by the penalty method (simple scalar penalty functions), where the algorithm is punished when there is a constraint violation. Similarly, the results are presented using best, worst, average, standard deviation (SD), and median performance indicators. The mean, standard deviation, and Wilcoxon signed-rank test were used to compare the different algorithms.

### 5.2.1 The welded beam design problem (WBD)

Table 10 shows the results of the GOA and nine other state-of-the-art algorithms used to compare the performance of the proposed algorithm. The GOA and SCA returned the

**Table 7** Wilcoxon signed-rank test results for classical test functions

| Pairwise comparison | | N | Mean Rank | Sum of ranks | Z | p-value |
|---|---|---|---|---|---|---|
| nAOA—GOA | Negative ranks | 0 | 0 | 0 | − 2.803[a] | 0.005 |
| | Positive ranks | 10 | 5.5 | 55 | | |
| | Ties | 5 | | | | |
| | Total | 15 | | | | |
| AOA—GOA | Negative ranks | 2 | 5.5 | 11 | − 1.682[a] | 0.093 |
| | Positive ranks | 8 | 5.5 | 44 | | |
| | Ties | 5 | | | | |
| | Total | 15 | | | | |
| CPSOGSA—GOA | Negative ranks | 1 | 13 | 13 | − 2.480[a] | 0.013 |
| | Positive ranks | 13 | 7.08 | 92 | | |
| | Ties | 1 | | | | |
| | Total | 15 | | | | |
| PSO—GOA | Negative ranks | 1 | 15 | 15 | − 2.556[a] | 0.011 |
| | Positive ranks | 14 | 7.5 | 105 | | |
| | Ties | 0 | | | | |
| | Total | 15 | | | | |
| BBO—GOA | Negative ranks | 4 | 9 | 36 | − 1.036[a] | 0.3 |
| | Positive ranks | 10 | 6.9 | 69 | | |
| | Ties | 1 | | | | |
| | Total | 15 | | | | |
| DE—GOA | Negative ranks | 0 | 0 | 0 | − 3.297[a] | 0.001 |
| | Positive ranks | 14 | 7.5 | 105 | | |
| | Ties | 0 | | | | |
| | Total | 14 | | | | |
| SSA—GOA | Negative ranks | 2 | 9.5 | 19 | − 2.103[a] | 0.035 |
| | Positive ranks | 12 | 7.17 | 86 | | |
| | Ties | 1 | | | | |
| | Total | 15 | | | | |
| SCA—GOA | Negative ranks | 0 | 0 | 0 | − 3.296[a] | 0.001 |
| | Positive ranks | 14 | 7.5 | 105 | | |
| | Ties | 1 | | | | |
| | Total | 15 | | | | |
| GWO—GOA | Negative ranks | 0 | 0 | 0 | − 2.666[a] | 0.008 |
| | Positive ranks | 9 | 5 | 45 | | |
| | Ties | 6 | | | | |
| | Total | 15 | | | | |

[a]Based on negative ranks

best cost value for the WBD. The GOA returned minimum values for mean and SD compared to other algorithms and optimal results for the design variables h, l, t, and b. The results showed the superiority and stability of the GOA, and good statistical results were obtained. The Wilcoxon signed-rank test null hypothesis is that "the mean distribution of the obtained results is the same." The mean comparison, the *p*-values, and Z scores of all the algorithms considered are shown in Table 11. It can be seen that all comparisons returned a positive rank except that with GWO. A negative rank means that the performance of GOA is inferior, while a positive rank implies otherwise. The result confirms that GOA outperformed all the algorithms used in the comparison, judging by the number of positive ranks returned, and is stable around the desired result. From Fig. 10, the convergence of GOA, nAOA, AOA, GWO, SCA, SSA, and CPSOGSA have similar behavior and lie close to each other. They converged steadily towards the optimal solution. The convergence curve for PSO, BBO, and DE confirms their sub-optimal results for the cost.

**Table 8** Friedman's test results for CEC 2020 test functions

| Algorithm | Mean rank | Rank |
|---|---|---|
| GOA | 1.20 | 1 |
| nAOA | 9.00 | 10 |
| AOA | 8.80 | 9 |
| CPSOGSA | 6.40 | 7 |
| PSO | 3.60 | 3 |
| BBO | 4.70 | 5 |
| DE | 3.25 | 2 |
| SSA | 4.55 | 4 |
| SCA | 7.60 | 8 |
| GWO | 5.90 | 6 |
| Test statistics | | |
| N | 10 | |
| Chi-Square | 62.465 | |
| df | 9 | |
| *P*-value | 0.000 | |

**Table 9** Wilcoxon signed-rank test results for CEC 2020 test functions

| Pairwise comparison | | N | Mean Rank | Sum of ranks | Z | p-value |
|---|---|---|---|---|---|---|
| nAOA—GOA | Negative ranks | 0 | .00 | .00 | − 2.803[a] | 0.005 |
| | Positive ranks | 10 | 5.50 | 55.00 | | |
| | Ties | 0 | | | | |
| | Total | 10 | | | | |
| AOA—GOA | Negative ranks | 0 | .00 | .00 | − 2.803[a] | 0.005 |
| | Positive ranks | 10 | 5.50 | 55.00 | | |
| | Ties | 0 | | | | |
| | Total | 10 | | | | |
| CPSOGSA—GOA | Negative ranks | 0 | .00 | .00 | − 2.803[a] | 0.005 |
| | Positive ranks | 10 | 5.50 | 55.00 | | |
| | Ties | 0 | | | | |
| | Total | 10 | | | | |
| PSO—GOA | Negative ranks | 2 | 1.50 | 3.00 | − 2.497[a] | 0.013 |
| | Positive ranks | 8 | 6.50 | 52.00 | | |
| | Ties | 0 | | | | |
| | Total | 10 | | | | |
| BBO—GOA | Negative ranks | 0 | .00 | .00 | − 2.803[a] | 0.005 |
| | Positive ranks | 10 | 5.50 | 55.00 | | |
| | Ties | 0 | | | | |
| | Total | 10 | | | | |
| DE—GOA | Negative ranks | 0 | .00 | .00 | − 2.803[a] | 0.005 |
| | Positive ranks | 10 | 5.50 | 55.00 | | |
| | Ties | 0 | | | | |
| | Total | 10 | | | | |
| SSA—GOA | Negative ranks | 0 | .00 | .00 | − 2.803[a] | 0.005 |
| | Positive ranks | 10 | 5.50 | 55.00 | | |
| | Ties | 0 | | | | |
| | Total | 10 | | | | |
| SCA—GOA | Negative ranks | 0 | .00 | .00 | − 2.803[a] | 0.005 |
| | Positive ranks | 10 | 5.50 | 55.00 | | |
| | Ties | 0 | | | | |
| | Total | 10 | | | | |
| GWO—GOA | Negative ranks | 0 | .00 | .00 | − 2.803[a] | 0.005 |
| | Positive ranks | 10 | 5.50 | 55.00 | | |
| | Ties | 0 | | | | |
| | Total | 10 | | | | |

[a]Based on negative ranks

**Table 10** Results for WBD

| | h | l | t | b | Best | Worst | Average | SD | Median |
|---|---|---|---|---|---|---|---|---|---|
| GOA | 0.181033 | 3.754998 | 9.036718 | 0.205729 | **1.6957** | 1.7468 | 1.7146 | 0.016293 | 1.7099 |
| nAOA | 0.126065 | 10 | 10 | 0.205654 | 1.8614 | 2.5375 | 2.3225 | 0.22526 | 2.4409 |
| AOA | 0.201542 | 3.470693 | 10 | 0.202481 | 1.7837 | 2.7743 | 2.117 | 0.26616 | 2.0254 |
| CPSOGSA | 0.194444 | 3.803513 | 7.956822 | 0.265357 | 1.6975 | 2.3115 | 1.8375 | 0.16261 | 1.7858 |
| PSO | 0.784094 | 2 | 2 | 2 | 1.09E + 14 | 1.09E + 14 | 1.09E + 14 | 0.047676 | 1.09E + 14 |
| BBO | 0.948771 | 1.632445 | 2 | 2 | 1.09E + 14 | 1.09E + 14 | 1.09E + 14 | 0.14866 | 1.09E + 14 |
| DE | 0.185461 | 3.654591 | 9.037761 | 0.205724 | 1.09E + 14 | 1.09E + 14 | 1.09E + 14 | 0.047676 | 1.09E + 14 |
| SSA | 0.181033 | 3.754998 | 9.036718 | 0.205729 | 1.6958 | 2.2096 | 1.7828 | 0.11086 | 1.739 |
| SCA | 0.142986 | 4.823892 | 9.17727 | 0.205437 | **1.6957** | 1.8961 | 1.8238 | 0.034276 | 1.8191 |
| GWO | 0.204848 | 3.273856 | 9.038443 | 0.205732 | 1.696 | 1.7024 | 1.6976 | 0.0014958 | 1.697 |

The algorithms with the best or superior solutions are given in bold

**Table 11** Wilcoxon signed ranks test statistics for WBD

| Pairwise comparison | | N | Mean Rank | Sum of ranks | Z | *p*-value |
|---|---|---|---|---|---|---|
| nAOA—GOA | Negative ranks | 2 | 1.50 | 3.00 | − 2.310[a] | 0.021 |
| | Positive ranks | 7 | 6.00 | 42.00 | | |
| | Ties | 0 | | | | |
| | Total | 9 | | | | |
| AOA—GOA | Negative ranks | 2 | 3.00 | 6.00 | − 1.955[a] | 0.051 |
| | Positive ranks | 7 | 5.57 | 39.00 | | |
| | Ties | 0 | | | | |
| | Total | 9 | | | | |
| CPSOGSA—GOA | Negative ranks | 1 | 9.00 | 9.00 | − 1.599[a] | 0.11 |
| | Positive ranks | 8 | 4.50 | 36.00 | | |
| | Ties | 0 | | | | |
| | Total | 9 | | | | |
| PSO—GOA | Negative ranks | 2 | 4.00 | 8.00 | − 1.733[a] | 0.083 |
| | Positive ranks | 7 | 5.29 | 37.00 | | |
| | Ties | 0 | | | | |
| | Total | 9 | | | | |
| BBO—GOA | Negative ranks | 2 | 4.50 | 9.00 | − 1.614[a] | 0.107 |
| | Positive ranks | 7 | 5.14 | 36.00 | | |
| | Ties | 0 | | | | |
| | Total | 9 | | | | |
| DE—GOA | Negative ranks | 2 | 3.00 | 6.00 | − 1.972[a] | 0.049 |
| | Positive ranks | 7 | 5.57 | 39.00 | | |
| | Ties | 0 | | | | |
| | Total | 9 | | | | |
| SSA—GOA | Negative ranks | 0 | .00 | .00 | − 2.023[a] | 0.043 |
| | Positive ranks | 5 | 3.00 | 15.00 | | |
| | Ties | 4 | | | | |
| | Total | 9 | | | | |
| SCA—GOA | Negative ranks | 2 | 2.00 | 4.00 | − 1.963[a] | 0.05 |
| | Positive ranks | 6 | 5.33 | 32.00 | | |
| | Ties | 1 | | | | |
| | Total | 9 | | | | |
| GWO—GOA | Negative ranks | 5 | 6.40 | 32.00 | − 1.125[b] | 0.26 |
| | Positive ranks | 4 | 3.25 | 13.00 | | |
| | Ties | 0 | | | | |
| | Total | 9 | | | | |

[a]Based on negative ranks

[b]Based on positive ranks

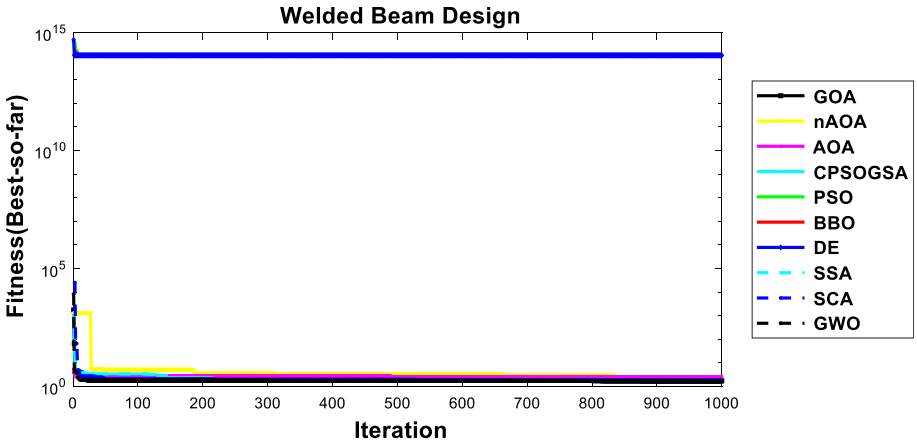


**Fig. 10** Convergence behavior of WBD

**Table 12** Result for CSD

| | d | D | N | Best | Worst | Average | SD | Median |
|---|---|---|---|---|---|---|---|---|
| GOA | 0.134406 | 1.16761 | 14.27422 | **3.6619** | 3.7124 | 3.6611 | 5.22E-06 | 3.6619 |
| nAOA | 0.149138 | 1.298134 | 15 | 3.931 | 8.0337 | 5.2189 | 1.0723 | 4.7877 |
| AOA | 0.147326 | 1.3 | 15 | 3.8481 | 7.1003 | 5.9769 | 0.7302 | 6.1513 |
| CPSOGSA | 0.134017 | 1.158641 | 14.44255 | 3.6619 | 3.7314 | 3.6764 | 0.024788 | 3.6619 |
| PSO | 2 | 1.2 | 2 | 409.77 | 409.77 | 409.77 | 2.89E-13 | 409.77 |
| BBO | 2 | 1.2 | 2 | 409.78 | 409.89 | 409.8 | 0.031362 | 409.79 |
| DE | 2 | 1.2 | 2 | 409.77 | 410.59 | 409.8 | 0.14961 | 409.77 |
| SSA | 0.134406 | 1.16761 | 14.27422 | **3.6619** | 3.7302 | 3.6795 | 0.017067 | 3.676 |
| SCA | 0.13948 | 1.278697 | 12.47937 | **3.6619** | 3.7399 | 3.6908 | 0.023965 | 3.6845 |
| GWO | 0.139127 | 1.3 | 11.88924 | **3.6619** | 3.6619 | 3.6619 | 5.23E-06 | 3.6619 |

The algorithms with the best or superior solutions are given in bold

**Table 13** Wilcoxon signed ranks test statistics for CSD

| Pairwise comparison | | N | Mean rank | Sum of ranks | *Z* | *p*-value |
|---|---|---|---|---|---|---|
| nAOA—GOA | Negative ranks | 0 | .00 | .00 | − 2.666[a] | 0.008 |
| | Positive ranks | 9 | 5.00 | 45.00 | | |
| | Ties | 0 | | | | |
| | Total | 9 | | | | |
| AOA—GOA | Negative ranks | 0 | .00 | .00 | − 2.666[a] | 0.008 |
| | Positive ranks | 9 | 5.00 | 45.00 | | |
| | Ties | 0 | | | | |
| | Total | 9 | | | | |
| CPSOGSA—GOA | Negative ranks | 2 | 1.50 | 3.00 | − 1.859[a] | 0.063 |
| | Positive ranks | 5 | 5.00 | 25.00 | | |
| | Ties | 2 | | | | |
| | Total | 9 | | | | |
| PSO—GOA | Negative ranks | 2 | 2.50 | 5.00 | − 2.075[a] | 0.038 |
| | Positive ranks | 7 | 5.71 | 40.00 | | |
| | Ties | 0 | | | | |
| | Total | 9 | | | | |
| BBO—GOA | Negative ranks | 1 | 4.00 | 4.00 | − 2.192[a] | 0.028 |
| | Positive ranks | 8 | 5.13 | 41.00 | | |
| | Ties | 0 | | | | |
| | Total | 9 | | | | |

**Table 13** (continued)

| Pairwise comparison | | N | Mean rank | Sum of ranks | Z | *p*-value |
|---|---|---|---|---|---|---|
| DE—GOA | Negative ranks | 1 | 4.00 | 4.00 | − 2.194[a] | 0.028 |
| | Positive ranks | 8 | 5.13 | 41.00 | | |
| | Ties | 0 | | | | |
| | Total | 9 | | | | |
| SSA—GOA | Negative ranks | 0 | .00 | .00 | − 2.023[a] | 0.043 |
| | Positive ranks | 5 | 3.00 | 15.00 | | |
| | Ties | 4 | | | | |
| | Total | 9 | | | | |
| SCA—GOA | Negative ranks | 1 | 8.00 | 8.00 | − 1.400[a] | 0.161 |
| | Positive ranks | 7 | 4.00 | 28.00 | | |
| | Ties | 1 | | | | |
| | Total | 9 | | | | |
| GWO—GOA | Negative ranks | 5 | 4.00 | 20.00 | − 1.014[b] | 0.31 |
| | Positive ranks | 2 | 4.00 | 8.00 | | |
| | Ties | 2[a] | | | | |
| | Total | 9 | | | | |

[a]Based on negative ranks

[b]Based on positive ranks

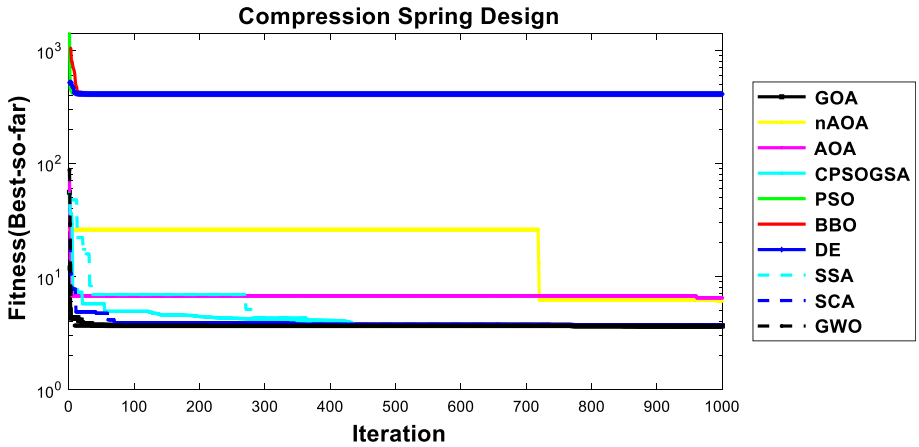


**Fig. 11** Convergence behavior of CSD

### 5.2.2 The compression spring design problem (CSD)

The results for CSD are shown in Table 12. It clearly shows that GOA, SSA, SCA, and GWO returned the best cost value; however, the GOA has the minimum values for mean, SD, and optimal results for the values for the design variables d, D, and N. This confirms the stability and superiority of GOA. The results of the Wilcoxon signed-rank test are shown in Table 13, and the comparison of the mean of GOA and all the algorithms considered returned a positive rank except that with GWO, the p-values and Z scores are shown in Table 13. A negative rank means that the performance of GOA is inferior, while a positive rank implies otherwise. The p-values indicate that the results of PSO, BBO, SSA, SCA, GWO, DE, and CPSOGSA are not statistically significant (*p*-values greater than the tolerance level of 0.05). The convergence shown in Fig. 11 confirms the respective performance of the algorithms used. The GOA, GWO, SCA, SSA, and CPSOGSA converged steadily towards the optimal solution. The nAOA and AOA are close to the optimal solution, whereas the PSO, BBO, and DE converged toward sub-optimal results for the cost.

### 5.2.3 The pressure vessel design problem (PVD).

The results shown in Table 14 show that the GOA returned the minimum value for the 'Best' indicator; the GOA outperforms all other algorithms in the 'Average' and 'SD'

**Table 14** Result for PVD

| | $T_s$ | $T_h$ | R | L | Best | Worst | Average | SD | Median |
|---|---|---|---|---|---|---|---|---|---|
| GOA | 0.547865 | 0.246925 | 43.49665 | 160.0482 | 4527.5 | 5106.1 | 4533.6 | 279.92 | 4702.6 |
| nAOA | 0.461132 | 0.240119 | 40.31962 | 200 | 4527.6 | 5527.6 | 4577.6 | 296.3 | 4727.3 |
| AOA | 0.376376 | 0.394446 | 40.64409 | 200 | 4739.7 | 6763.7 | 5592.3 | 549.35 | 4902 |
| CPSOGSA | 1.093574 | 10 | 65.22523 | 10 | 5302.6 | 7638.3 | 5926.5 | 524.32 | 6624.3 |
| PSO | 10 | 10 | 53.7325 | 71.23984 | 2.04E + 05 | 2.04E + 05 | 2.04E + 05 | 0.70492 | 2.04E + 05 |
| BBO | 10 | 10 | 57.44469 | 48.42015 | 2.04E + 05 | 2.12E + 05 | 2.06E + 05 | 2100.2 | 2.05E + 05 |
| DE | 10 | 10 | 53.52146 | 72.6505 | 2.04E + 05 | 2.04E + 05 | 2.04E + 05 | 0.18563 | 2.04E + 05 |
| SSA | 0.472805 | 0.02345 | 40.735 | 194.2972 | 5302.6 | 6638.5 | 5658.6 | 502.02 | 5668.4 |
| SCA | 10 | 0.5432 | 40.32316 | 200 | 5302.6 | 6062.5 | 5435.3 | 1414.6 | 5656.2 |
| GWO | 1.09575 | 0.02134 | 65.226 | 10 | 5302.6 | 6063.2 | 5680 | 1145.6 | 5702.8 |

**Table 15** Wilcoxon signed ranks test statistics for PVD

| Pairwise comparison | | *N* | Mean Rank | Sum of ranks | *Z* | *p*-value |
|---|---|---|---|---|---|---|
| nAOA—GOA | Negative Ranks | 1 | 3.00 | 3.00 | − 2.100[a] | 0.036 |
| | Positive Ranks | 7 | 4.71 | 33.00 | | |
| | Ties | 1 | | | | |
| | Total | 9 | | | | |
| AOA—GOA | Negative Ranks | 1 | 2.00 | 2.00 | − 2.429[a] | 0.015 |
| | Positive Ranks | 8 | 5.38 | 43.00 | | |
| | Ties | 0 | | | | |
| | Total | 9 | | | | |
| CPSOGSA—GOA | Negative Ranks | 2 | 3.50 | 7.00 | − 1.540[a] | 0.123 |
| | Positive Ranks | 6 | 4.83 | 29.00 | | |
| | Ties | 1 | | | | |
| | Total | 9 | | | | |
| PSO—GOA | Negative Ranks | 2 | 4.50 | 9.00 | − 1.601[a] | 0.109 |
| | Positive Ranks | 7 | 5.14 | 36.00 | | |
| | Ties | 0 | | | | |
| | Total | 9 | | | | |
| BBO—GOA | Negative Ranks | 1 | 4.00 | 4.00 | − 2.192[a] | 0.028 |
| | Positive Ranks | 8 | 5.13 | 41.00 | | |
| | Ties | 0 | | | | |
| | Total | 9 | | | | |
| DE—GOA | Negative Ranks | 2 | 4.50 | 9.00 | − 1.601[a] | 0.109 |
| | Positive Ranks | 7 | 5.14 | 36.00 | | |
| | Ties | 0 | | | | |
| | Total | 9 | | | | |
| SSA—GOA | Negative Ranks | 1 | 1.00 | 1.00 | − 1.753[a] | 0.08 |
| | Positive Ranks | 4 | 3.50 | 14.00 | | |
| | Ties | 4 | | | | |
| | Total | 9 | | | | |

**Table 15** (continued)

| Pairwise comparison | | *N* | Mean Rank | Sum of ranks | *Z* | *p*-value |
|---|---|---|---|---|---|---|
| SCA—GOA | Negative Ranks | 2 | 1.50 | 3.00 | − 1.859[a] | 0.063 |
| | Positive Ranks | 5 | 5.00 | 25.00 | | |
| | Ties | 2 | | | | |
| | Total | 9 | | | | |
| GWO—GOA | Negative Ranks | 1 | 5.00 | 5.00 | − 1.521[a] | 0.128 |
| | Positive Ranks | 6 | 3.83 | 23.00 | | |
| | Ties | 2 | | | | |
| | Total | 9 | | | | |

[a]Based on negative ranks

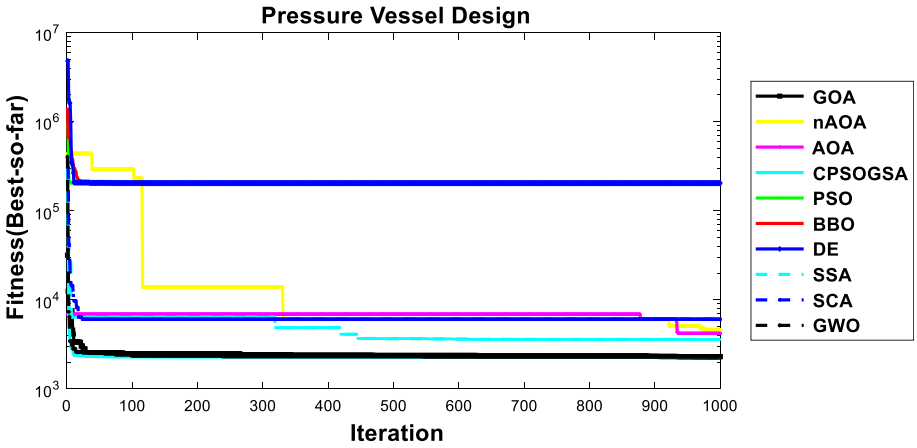


**Fig. 12** Convergence behavior of PVD

performance indicators. The Wilcoxon signed-rank test (Table 15) returned a positive rank and p-values less than 0.05 for the mean comparison of GOA with SSA, nAOA, AOA, DE, PSO, and BBO, which indicates that results with GOA having the better performance are significant. This performance is attributed to the convergence of the GOA towards the optimal value. Similar observations can be made on the convergence behavior of PVD shown in Fig. 12. It confirms the performance of the respective algorithms discussed earlier.

#### 5.2.4 The speed reducer design problem (SRD)

Table 16 shows the results of the algorithms used to compare the performance of the GOA. The GOA and CPSOGSA returned a very competitive result for the 'Best' cost value for the SRD; however, CPSOGSA converged quickly towards the best results, evident from the values returned by 'SD and Median' for CPSOGSA. GOA returned minimum values for the design variables $x_1, x_2, x_3, x_4, x_5, x_6, x_7$. The results showed the competitiveness and stability of the GOA, and good statistical results were obtained. The comparison of the mean of GOA and all the algorithms considered returned a positive rank except that with CPSOGSA, the p-values and Z scores are shown in Table 17. A negative rank means that the performance of GOA is inferior, while a positive rank implies otherwise. The result confirms that GOA returned a better performance and is stable around the desired result. Figure 13 shows that the convergence of GOA, nAOA, AOA, GWO, SCA, SSA, and CPSOGSA have similar behavior and lie close to each other. They converged steadily towards the optimal solution. The convergence curve for PSO, BBO, and DE confirms their sub-optimal results for the cost.

## 6 Conclusion and future work

### 6.1 Conclusion

This paper proposed a new metaheuristic algorithm called the Gazelle Optimization Algorithm (GOA), inspired by the gazelles' survival ability in their predator-dominated environment. Every day, the gazelle knows that if it does not outrun and outmaneuver its predators, it becomes meat for the day. Despite being down in the food chain, they are not classified as endangered, which means they are doing

**Table 16** Result for SRD

| | × 1 | × 2 | × 3 | × 4 | × 5 | × 6 | × 7 | Best | Worst | Average | SD | Median |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GOA | 3.5 | 0.7 | 17 | 7.3 | 7.7 | 3.349371 | 5.287884 | **2994.1** | 3007.9 | 2998.2 | 3.7715 | 2996.9 |
| nAOA | 3.5 | 0.7 | 17 | 7.3 | 8.3 | 3.570621 | 5.314766 | 3065.7 | 3214.8 | 3135.5 | 47.609 | 3114.6 |
| AOA | 3.6 | 0.7 | 17 | 7.3 | 8.3 | 3.436436 | 5.361661 | 3073.4 | 3224 | 3134.1 | 42.143 | 3122.1 |
| CPSOGSA | 3.5 | 0.7 | 17 | 7.3 | 7.7 | 3.350056 | 5.285631 | **2993.6** | 2993.6 | 2993.6 | 5.47E-13 | 2993.6 |
| PSO | 5.5 | 5 | 5 | 5.5 | 5.5 | 5 | 5.101362 | 1.32E + 07 | 1.32E + 07 | 1.32E + 07 | 4.02E-09 | 1.32E + 07 |
| BBO | 5.5 | 5 | 5 | 5.5 | 5.5 | 5 | 5.101362 | 1.32E + 07 | 1.32E + 07 | 1.32E + 07 | 3.79E-09 | 1.32E + 07 |
| DE | 5.5 | 5 | 5 | 5.5 | 5.5 | 5 | 5.101362 | 1.32E + 07 | 1.32E + 07 | 1.32E + 07 | 4.24E-09 | 1.32E + 07 |
| SSA | 3.5 | 0.7 | 17 | 7.6 | 7.7 | 3.438005 | 5.285632 | 2999.3 | 3068.4 | 3018.6 | 19.416 | 3010.7 |
| SCA | 3.6 | 0.7 | 17 | 8.2 | 8.3 | 3.548614 | 5.291943 | 2994.1 | 3160.3 | 3098.3 | 28.036 | 3091.5 |
| GWO | 3.5 | 0.7 | 17 | 7.5 | 8.3 | 3.349558 | 5.285315 | 2995.7 | 3008.2 | 3001 | 3.349 | 3000.8 |

The algorithms with the best or superior solutions are given in bold

something right. One of those right things the gazelles do is to escape from predators. However, studies show that the predators are successful only 34% time. This study used this information to develop a new metaheuristic algorithm that uses the gazelle's survival abilities to solve real-world problems.

The newly developed algorithm was modeled as a population-based metaheuristic algorithm whose initial population is stochastically generated. The optimization process is two-phased: the exploitation phase of the algorithm simulates the gazelles grazing peacefully in the absence of the predator or while the predator is stalking it. The GOA goes into the exploration phase once a predator is spotted, it consists of the gazelle outrunning and outmaneuvering the predator to a haven. These phases are repeated subject to termination criteria to find optimal solutions to optimization problems. Each phase is modeled mathematically by different equations and tested using 25 benchmark functions (15 classical and 10 composited functions) and three (3) engineering design problems.

The simulation results confirm the superiority and competitiveness of the proposed algorithm over nine state-of-the-art algorithms available in the literature. These algorithms are candidate representations of different categories of metaheuristic algorithms available in the literature. Five different performance metrics were used to report the results. Also, the results are statistically compared using Friedman's Two-Way Analysis of Variance by Ranks and the Wilcoxon test. The GOA returned optimal solutions for most of the problems considered and was very competitive in the remaining function. Standard statistical results also confirmed the superiority of the GOA over the selected state-of-the-art algorithms considered. Finally, the results showed that the proposed GOA could solve most problems, perform better than the previous algorithms, and compete with better methods.

## 6.2 Future work

For future research direction, researchers can try to overcome the drawback of premature convergence observed in F5 and F6. Researchers could also consider hybridizing the GOA with other algorithms or using chaotic maps with properties like stochasticity, ergodicity, and complex nonlinear motion to overcome this drawback. In addition, GOA could be applied to many other real-world problems, such as parallel machine scheduling problems, economic load dispatch problems of electronic science, and medical image processing.

**Table 17** Wilcoxon signed ranks test statistics for SRD

| Pairwise comparison | | *N* | Mean Rank | Sum of Ranks | *Z* | *p*-value |
|---|---|---|---|---|---|---|
| nAOA—GOA | Negative Ranks | 1 | 1.00 | 1.00 | − 2.701[a] | 0.007 |
| | Positive Ranks | 9 | 6.00 | 54.00 | | |
| | Ties | 2 | | | | |
| | Total | 12 | | | | |
| AOA—GOA | Negative Ranks | 1 | 1.00 | 1.00 | − 2.701[a] | 0.007 |
| | Positive Ranks | 9 | 6.00 | 54.00 | | |
| | Ties | 2 | | | | |
| | Total | 12 | | | | |
| CPSOGSA—GOA | Negative Ranks | 9 | 5.78 | 52.00 | − 2.497[b] | 0.013 |
| | Positive Ranks | 1 | 3.00 | 3.00 | | |
| | Ties | 2 | | | | |
| | Total | 12 | | | | |
| PSO—GOA | Negative Ranks | 5 | 4.60 | 23.00 | − 1.255[a] | 0.209 |
| | Positive Ranks | 7 | 7.86 | 55.00 | | |
| | Ties | 0 | | | | |
| | Total | 12 | | | | |
| BBO—GOA | Negative Ranks | 5 | 4.60 | 23.00 | − 1.255[a] | 0.209 |
| | Positive Ranks | 7 | 7.86 | 55.00 | | |
| | Ties | 0 | | | | |
| | Total | 12 | | | | |
| DE—GOA | Negative Ranks | 5 | 4.60 | 23.00 | − 1.255[a] | 0.209 |
| | Positive Ranks | 7 | 7.86 | 55.00 | | |
| | Ties | 0 | | | | |
| | Total | 12 | | | | |
| SSA—GOA | Negative Ranks | 4 | 2.50 | 10.00 | − 2.045[a] | 0.041 |
| | Positive Ranks | 7 | 8.00 | 56.00 | | |
| | Ties | 1 | | | | |
| | Total | 12 | | | | |
| SCA—GOA | Negative Ranks | 1 | 1.00 | 1.00 | − 2.701[a] | 0.007 |
| | Positive Ranks | 9 | 6.00 | 54.00 | | |
| | Ties | 2 | | | | |
| | Total | 12 | | | | |
| GWO—GOA | Negative Ranks | 4 | 3.75 | 15.00 | − 1.600[a] | 0.11 |
| | Positive ranks | 7 | 7.29 | 51.00 | | |
| | Ties | 1 | | | | |
| | Total | 12 | | | | |

[a]Based on negative ranks

[b]Based on positive ranks

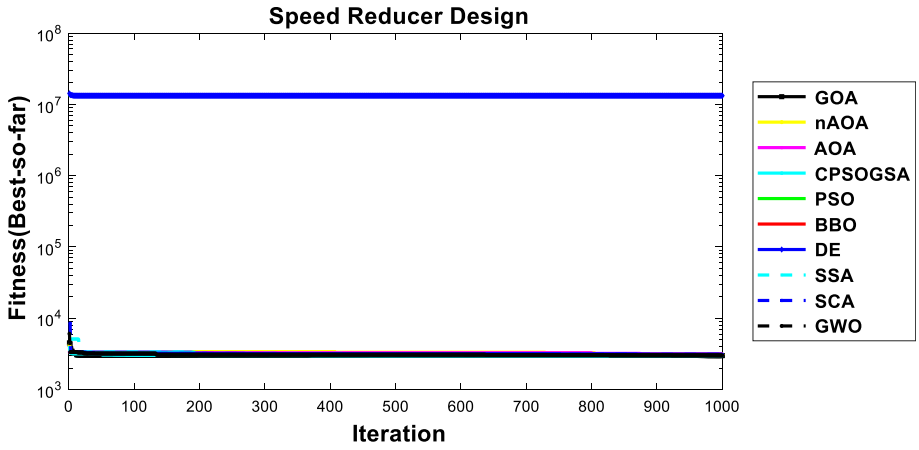


**Fig. 13** Convergence behavior of SRD

**Data availability statements** Data is available from the authors upon reasonable request.

## Declarations

**Conflict of Interest** The authors declare that there is no conflict of interest regarding the publication of this paper.

**Ethical approval** This article does not contain any studies with human participants or animals performed by any of the authors.

**Informed consent** Informed consent was obtained from all individual participants included in the study.

## References

1. Abed-alguni BH, Paul D (2022) Island-based Cuckoo Search with elite opposition-based learning and multiple mutation methods for solving optimization problems. Soft Comput 26(7):3293–3312
2. Abed-Alguni BH, Paul D, Hammad R (2022) Improved Salp swarm algorithm for solving single-objective continuous optimization problems. Appl Intell. https://doi.org/10.1007/s10489-022-03269-x
3. Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi AH (2021) The arithmetic optimization algorithm. Comput Methods Appl Mech Eng 376:113609
4. Agushaka JO, Ezugwu AE (2020). Diabetes classification techniques: a brief state-of-the-art literature review. In: International Conference on Applied Informatics (pp. 313–329). Ogun: Springer, Cham
5. Agushaka JO, Ezugwu AE (2021) Advanced arithmetic optimization algorithm for solving mechanical engineering design problems. PLoS ONE. https://doi.org/10.1371/journal.pone.0255703
6. Agushaka JO, Ezugwu AE (2022) Influence of probability distribution initialization methods on the performance of advanced arithmetic optimization algorithm with application to unrelated parallel machine scheduling problem. Concurr Comput Pract Exper 34:e6871
7. Agushaka JO, Ezugwu AE, Abualigah L (2022) Dwarf Mongoose Optimization Algorithm. Comput Methods Appl Mech Eng 391:114570
8. Agushaka J, Ezugwu A (2020) Influence of initializing krill herd algorithm with low-discrepancy sequences. IEEE Access 8:210886–210909
9. Akay B, Karaboga D (2012) A modified artificial bee colony algorithm for real-parameter optimization. Inf Sci 192:120–142
10. Akay B, Karaboga D (2012) Artificial bee colony algorithm for large-scale problems and engineering design optimization. J Intell Manuf 23(4):1001–1014
11. Alatas B (2011) ACROA: artificial chemical reaction optimization algorithm for global optimization. Expert Syst Appl 38(10):13170–13180
12. Arora S, Singh S (2019) Butterfly optimization algorithm: a novel approach for global optimization. Soft Comput 23(3):715–734
13. Atashpaz-Gargari E, Lucas C (2007). Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In 2007 IEEE congress on evolutionary computation (pp. 4661–4667). Ieee
14. Biswas A, Mishra K, Tiwari S, Misra A (2013). Physics-inspired optimization algorithms: a survey. Journal of Optimization, 2013
15. Braik MS (2021) Chameleon swarm algorithm: a bio-inspired optimizer for solving engineering design problems. Expert Syst Appl 174:114685
16. Coello C (2000) Use of self-adaptive penalty approach for engineering optimization problems. Comput Ind 41(2):113–127
17. Dorigo, M., & Di Caro, G. (1999). Ant colony optimization: a new meta-heuristic. In: Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406) (Vol. 2) (pp. 1470–1477). IEEE
18. Einstein A (1956) Investigations on the Theory of the Brownian Movement. Courier Corporation, US
19. Estes R (2020, May 2). Gazelle. Retrieved from Encyclopedia Britannica: https://www.britannica.com/animal/gazelle
20. Ezugwu AE (2020) Nature-inspired metaheuristic techniques for automatic clustering: a survey and performance study. SN Applied Sciences 2(2):273
21. Ezugwu AE, Adeleke OJ, Akinyelu AA, Viriri S (2020) A conceptual comparison of several metaheuristic algorithms on continuous optimization problems. Neural Comput Appl 32(10):6207–6251
22. Ezugwu AE, Akutsah F (2018) An improved firefly algorithm for the unrelated parallel machines scheduling problem with sequence-dependent setup times. IEEE Access 6:54459–54478
23. Ezugwu AE, Agushaka JO, Abualigah L, Mirjalili S, Gandomi AH (2022) Prairie dog optimization algorithm. Neural Comput Appl. https://doi.org/10.1007/s00521-022-07530-9
24. Ezugwu AE, Shukla AK, Nath R, Akinyelu AA, Agushaka JO, Chiroma H, Muhuri PK (2021) Metaheuristics: a comprehensive

overview and classification along with bibliometric analysis. Artif Intell Rev 54(6):4237–4316
25. Faramarzi A, Heidarinejad M, Mirjalili S, Gandomi AH (2020) Marine predators algorithm: a nature-inspired metaheuristic. Expert Syst Appl 152:113377
26. FitzGibbon, C. D., & Lazarus, J. (1995). Antipredator behavior of Serengeti ungulates: individual differences and population consequences. Serengeti II: Dynamics, management, and conservation of an ecosystem, 274–296
27. Hassan BA (2021) CSCF: a chaotic sine cosine firefly algorithm for practical application problems. Neural Comput Appl 33(12):7011–7030
28. Holland, J. H. (1975). Adaptation in natural and artificial systems. University of Michigan Press. (Second. Michigan: University of Michigan Press. (Second edition: MIT Press, 1992)
29. Houssein EH, Saad MR, Hashim FA, Shaban H, Hassaballah M (2020) Lévy flight distribution: a new metaheuristic algorithm for solving engineering optimization problems. Eng Appl Artif Intell 94:103731
30. Humphries NE, Queiroz N, Dyer JR, Pade NG, Musyl MK, Schaefer KM, Sims DW (2010) Environmental context explains Lévy and Brownian movement patterns of marine predators. Nature 465(7301):1066–1069
31. Ibrahim A, Tawhid M, Ward R (2020) A binary water wave optimization for feature selection. Int J Approx Reason 120:74–91
32. Kaveh A, Eslamlou A (2020) Water strider algorithm: a new metaheuristic and applications. Structures 25:520–541
33. Kaveh A, Hamedani KB, Kamalinejad M (2022) Improved slime mould algorithm with elitist strategy and its application to structural optimization with natural frequency constraints. Comput Struct 264:106760
34. Kaveh A, Talatahari S, Khodadadi N (2020). Stochastic paint optimizer: theory and application in civil engineering. Engineering with Computers, 1–32
35. Kennedy J, Eberhart R (1995). Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks (Vol. 4) (pp. 1942–1948). IEEE
36. Kirkpatrick S, Gelatt Jr CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680
37. Lee KS, Geem ZW (2005) A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comput Methods Appl Mech Eng 194(36–38):3902–3933
38. Liu H, Zhang X, Liang H, Tu L (2020) Stability analysis of the human behavior-based particle swarm optimization without stagnation assumption. ExpertSystAppl 159:113638
39. Mantegna RN (1994) Fast, accurate algorithm for numerical simulation of Levy stable stochastic processes. Phys Rev E 49(5):4677
40. Mezura-Montes E, Coello CA (2005). Useful infeasible solutions in engineering optimization with evolutionary algorithms. In: Mexican international conference on artificial intelligence (pp. 652–662). Berlin, Heidelberg: Springer
41. Mirjalili S (2016) SCA: a sine cosine algorithm for solving optimization problems. Knowl-Based Syst 96:120–133
42. Mirjalili S, Gandomi A, Mirjalili S, Saremi S, Faris H, Mirjalili S (2017) Salp swarm algorithm: a bioinspired optimizer for engineering design problems. Adv Eng Software 124:163–191
43. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
44. Mohammadi F, Amini M, Arabnia H (2020). Evolutionary computation, optimization, and learning algorithms for data science. Optimization, Learning, and Control for Interdependent Complex Networks.Cham, Switzerland: Springer, 37–65
45. Moosavi S, Bardsiri V (2019) Poor and rich optimization algorithm: a new human-based and multi populations algorithm. Eng Appl Artif Intel 86:165–181
46. Nadimi-Shahraki MH, Taghian S, Mirjalili S (2021) An improved grey wolf optimizer for solving engineering problems. Expert Syst Appl 166:113917
47. Olson KA, Larsen EA, Mueller T, Leimgruber P, Fuller TK, Schaller GB, Fagan WF (2014) Survival probabilities of adult Mongolian gazelles. J Wildl Manag 78(1):35–41
48. Omondi, S. (2017, August 1). Gazelle Facts - Animals of the World. Retrieved from WorldAtlas/Environment: https://www.worldatlas.com/articles/gazelle-facts-animals-of-the-world.html
49. Oyelade ON, Ezugwu AE-S, Mohamed TI, Abualigah L (2022) Ebola optimization search algorithm: A new nature-inspired metaheuristic optimization algorithm IEEE. Access 10:16150–16177
50. Pelusi D, Mascella R, Tallini L, Nayak J, Naik B, Deng Y (2020) Improving exploration and exploitation via a hyperbolic gravitational search algorithm. Knowl-Based Syst 193:105
51. Rahkar Farshi T (2021) Battle royale optimization algorithm. Neural Comput Appl 33(4):1139–1157
52. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179(13):2232–2248
53. Rather S, Bala P (2019). Hybridization of constriction coefficient based particle swarm optimization and gravitational search algorithm for function optimization. In: International Conference on Advances in Electronics, Electrical, and Computational Intelligence (ICAEEC- 2019). Elsevier
54. Rechenberg I (1978). Evolutionary strategies. In simulation methods in medicine and biology, Berlin, Heidelberg, 83–114
55. Sarzaeim P, Bozorg-Haddad O, Chu X (2018). Teaching-learning-based optimization (TLBO) algorithm. Advanced Optimization by Nature-Inspired Algorithms. Singapore, Asia: Springer, 51–58
56. Shabani A, Asgarian B, Salido M, Gharebaghi SA (2020) Search and rescue optimization algorithm: a new optimization method for solving constrained engineering optimization problems. Expert Syst Appl 161:113698
57. Simon D (2008) Biogeography based optimization. IEEE Trans Evol Comput 12(6):702–713
58. Slowik A, Kwasnicka H (2020) Evolutionary algorithms and their applications to engineering problems. Neural Comput Appl 32(16):12363–12379
59. Storn R, Price K (1997) Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 11(4):341–359
60. Tzanetos A, Dounias G (2021) Nature inspired optimization algorithms or simply variations of metaheuristics? Artif Intell Rev 54(3):1841–1862
61. Uymaz SA, Tezel G, Yel E (2015) Artificial algae algorithm (AAA) for nonlinear global optimization. Appl Soft Comput 31:153–171
62. Wang GG, Guo L, Gandomi AH, Hao GS, Wang H (2014) Chaotic krill herd algorithm. Inf Sci 274:17–34
63. Xie Q, Cheng G, Zhang X, Peng L (2020) Feature selection using improved forest optimization algorithm. Inf Technol Control 49(2):289–301
64. Xing B, Gao W (2014). Invasive weed optimization algorithm. Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms. Cham, Switzerland:Springer, 177–181
65. Yang X S, Deb S (2009). Cuckoo search via Lévy flights. In 2009 World congress on nature & biologically inspired computing (NaBIC) (pp. 210–214). Ieee
66. Yang X, Karamanoglu M (2020). Nature-inspired computation and swarm intelligence: a state-of-the-art overview. Nature-

Inspired Computation and Swarm Intelligence. Cambridge, Massachusetts: Academic Press, 3–18
67. Zamani H, Nadimi-Shahraki MH, Gandomi AH (2022) Starling murmuration optimizer: a novel bio-inspired algorithm for global and engineering optimization. Comput Methods Appl Mech Eng 392:114616
68. Zhang P, Wang C, Qin Z, Cao H (2022) A multidomain virtual network embedding algorithm based on multiobjective optimization for Internet of Drones architecture in Industry 40. Softw Pract Exper 52(3):710–728
69. Zhao W, Du C, Jiang S (2018) An adaptive multiscale approach for identifying multiple flaws based on XFEM and a discrete artificial fish swarm algorithm. Comput Methods Appl Mech Eng 339:341–357
70. Zhao W, Wang L, Mirjalili S (2022) Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications. Comput Methods Appl Mech Eng 388:114194