Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 391 (2022) 114570
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Dwarf Mongoose Optimization Algorithm

Jeffrey O. Agushaka[a], Absalom E. Ezugwu[a], Laith Abualigah[b,c,*]

[a] School of Mathematics, Statistics, and Computer Science, University of KwaZulu-Natal, King Edward Road, Pietermaritzburg, KwaZulu-Natal 3201, South Africa
[b] Faculty of Computer Sciences and Informatics, Amman Arab University, Amman 11953, Jordan
[c] School of Computer Sciences, Universiti Sains Malaysia, Pulau Pinang 11800, Malaysia

Received 17 December 2021; received in revised form 31 December 2021; accepted 31 December 2021
Available online 28 January 2022

Abstract

This paper proposes a new metaheuristic algorithm called dwarf mongoose optimization algorithm (DMO) to solve the classical and CEC 2020 benchmark functions and 12 continuous/discrete engineering optimization problems. The DMO mimics the foraging behavior of the dwarf mongoose. The restrictive mode of prey capture (feeding) has dramatically affected the mongooses' social behavior and ecological adaptations to compensate for efficient family nutrition. The compensatory behavioral adaptations of the mongoose are prey size, space utilization, group size, and food provisioning. Three social groups of the dwarf mongoose are used in the proposed algorithm, the alpha group, babysitters, and the scout group. The family forage as a unit, and the alpha female initiates foraging, determines the foraging path, the distance covered, and the sleeping mounds. A certain number of the mongoose population (usually a mixture of males and females) serve as the babysitters. They remain with the young until the group returns at midday or evening. The babysitters are exchanged for the first to forage with the group (exploitation phase). The dwarf mongooses do not build a nest for their young; they move them from one sleeping mound to another and do not return to the previously foraged site. The dwarf mongoose has adopted a seminomadic way of life in a territory large enough to support the entire group (exploration phase). The nomadic behavior prevents overexploitation of a particular area. It also ensures exploration of the whole territory because no previously visited sleeping mound is returned. The performance of the proposed DMO algorithm is compared with seven other algorithms to show its effectiveness in terms of different performance metrics and statistics. In most cases, the near-optimal solutions achieved by the DMO are better than the best solutions obtained by the current state-of-the-art algorithms. Matlab codes of DMO are available at https://www.mathworks.com/matlabcentral/fileexchange/105125-dwarf-mongoose-optimization-algorithm.
© 2022 Elsevier B.V. All rights reserved.

Keywords: Dwarf Mongoose Optimization Algorithm; Metaheuristic; Nature-inspired algorithms; Global optimization; Engineering design problems

1. Introduction

Nature has been a source of inspiration for many metaheuristic algorithms. Researchers have successfully used natural phenomena to develop metaheuristic algorithms [1]. For instance, natural selection in the theory of evolutions is the inspiration behind the Genetic Algorithm (GA) [2]. The apparent swarm intelligence of bird flocks is the

* Corresponding author at: Faculty of Computer Sciences and Informatics, Amman Arab University, Amman 11953, Jordan.
E-mail addresses: 208088307@stu.ukzn.ac.za (J.O. Agushaka), Ezugwua@ukzn.ac.za (A.E. Ezugwu), aligah.2020@gmail.com (L. Abualigah).

https://doi.org/10.1016/j.cma.2022.114570
0045-7825/© 2022 Elsevier B.V. All rights reserved.

inspiration behind particle swarm optimization (PSO) [3]. Generally, nature-inspired metaheuristic algorithms have successfully solved various domains, including the traveling salesman problem [4], optimal control [5], medical image processing [6], and others [7,8]. The success of nature-inspired algorithms stems from imitating the best aspects of nature.

Optimization occurs in every facet of our life, ranging from medicine, engineering, manufacturing, and many more. Two well-known methods of solving optimization problems are the mathematical and metaheuristic approach. The mathematical methods are gradient-dependent, sensitive to the initial starting point [9]. This drawback and the nature and complexity of global optimization problems have triggered a surge in the rate at which nature-inspired algorithms are proposed [10]. Their reliance on a metaphor-based paradigm has generated criticisms [11]. However, there have been many success stories, especially in finding solutions to complex optimization problems in benchmarks like CEC 2020 [12] and real-world engineering scenarios [13].

Different metaheuristic algorithms behave differently when solving problems depending on the natural phenomena they mimic. No one algorithm solves all problems optimally, hence the need to develop a new high-performance algorithm that solves specific types of problems. Relying on the no-free lunch theory, novel metaheuristic algorithms have been developed regularly to find better optimal solutions for particular complex and large-scale optimization problems. Most researchers focus on developing algorithms that better search the problem space and balance the exploration and exploitation of the proposed algorithm [14]. The last decade has seen an exponential increase in the number of nature-inspired optimization algorithms. The authors claim inspiration from nature, novelty, and strong optimization capability.

Although many algorithms are inspired by foraging and social-behavioral structure in nature [15–18], the unique compensatory behavioral adaptations of the dwarf mongoose have not been modeled to solve optimization problems. These unique adaptations make the dwarf mongoose stand out as the most petite carnivore and give the DMO a competitive advantage over some state-of-the-art algorithms used in this study. Therefore, in this study, we present a novel metaheuristic algorithm that mimics the foraging behavior of the dwarf mongoose using their compensatory behavioral adaptations. We modeled three social structural transformations for the optimization process: the alpha group, babysitters, and the scout group. The optimization starts with the alpha group setting out (exploring space) foraging, leaving the babysitters and the young at the nest.

Once a foraging site is found, the alpha group feeds until midday, simulated by the babysitter exchange criteria, when they return to exchange babysitters. Once the babysitters are exchanged, they do not return to the previously foraged site to avoid overgrazing. The scouts would have identified a new foraging site and informed the alpha female, leading the family to the new site. The babysitter exchange sets out a new exploration phase, followed by intensive exploitation till evening when the clan return to a new sleeping nest. The proposed algorithm is used to solve benchmark test functions and 12 different optimization problems in the engineering domain. The obtained results of DMO are compared with seven(7) other optimizers in terms of five(5) performance indicators, and in most cases, the solutions of the DMO are better than the other methods.

The proposed DMO theoretically can find the global optimum solutions of different optimization problems better than some algorithms in literature because of the following unique attributes:

- The DMO stochastically creates and improves a set of candidate solutions for a given optimization problem, relying on the exploratory and exploitation ability of DMO, which mimics the seminomadic behavior and compensatory adaptation of the dwarf mongoose.
- Different regions of the problem search space are explored as the dwarf mongoose move from one food source or sleeping mound to another
- Promising regions of the search space are exploited because the DMO is modeled after the dwarf mongoose's inability to capture large prey for family feeding but individually sourcing enough food to satisfy the individual.
- The DMO has only one parameter that can be tuned.

The rest of the paper is organized as follows: Section 2 gives a brief background on the different proposed nature-inspired algorithms. In Section 3, the proposed dwarf mongoose optimization algorithm (DMO). Section 4 presents the experimental setup, results, and detailed discussion. Finally, the conclusion and future work is presented in Section 5.

2. Background

The last decade has seen a tremendous increase in the rate of new proposed nature-inspired metaheuristic algorithms. Different natural phenomena have inspired many state-of-the-art algorithms, including biological,

Table 1
Some nature-inspired metaheuristic algorithms (2019–2021).

Algorithm	Inspiration	Reference
Group teaching optimization algorithm	Group teaching mechanism	[35]
Black widow optimization algorithm	Unique mating behavior of black widow spiders.	[36]
Chaos Game Optimization	Some principles of chaos theory	[37]
Adolescent Identity Search Algorithm (AISA)	Process of identity development/search of adolescents	[38]
Atomic orbital search	Basic principles of quantum mechanics	[39]
A novel metaheuristic optimizer inspired by behavior of jellyfish in the ocean	Behavior of jellyfish in the ocean	[40]
Quantum dolphin swarm algorithm	Dolphin swarm algorithm	[41]
Arithmetic optimization algorithm	Arithmetic operators	[42]
Advanced arithmetic optimization algorithm	Advanced arithmetic operators	[43]
Ebola Optimization Search Algorithm (EOSA)	Ebola virus	[6]
Golden ratio optimization method (GROM)	Growth in nature using the golden ratio of Fibonacci series	[44]
Bald eagle search optimization algorithm	Bald eagle	[45]
Black Hole Mechanics Optimization	Mechanics of black holes	[46]
Capuchin search algorithm	Capuchin monkeys	[47]
Tiki-taka algorithm	Football playing style	[48]
Cooperation search algorithm	Team cooperation behaviors in modern enterprise	[49]
Aquila Optimizer	Aquila bird	[50]
The Sailfish Optimizer	The Sailfish group hunting	[51]
Social Network Search	Social network user's efforts to gain more popularity	[52]

physics-based, chemistry-based, and many more. The proposed algorithms have been used to solve different optimization problems with varying degrees of success [19–27]. It is equally important to note that many of these algorithms have not been tested on all complex optimization problems or applied to handle all real-world optimization problems.

The success of these proposed algorithms has drawn tremendous interest from researchers with great effort to improve their performances. An advanced discrete firefly algorithm was used for scheduling unrelated parallel machines with sequence-dependent setup times [28]. Characterization of abnormalities in breast cancer images was achieved using a nature-inspired metaheuristic optimized convolutional neural networks model [29]. A performance study of different metaheuristic approaches for solving the quadratic assignment problem was presented in [30]. The symbiotic organism search algorithm was boosted with ecosystem service for dynamic blood allocation in the blood banking system [31].

Some example of newly proposed metaheuristic algorithms within the last two years (2019–2021) is shown in Table 1. Unfortunately, we see that some of the inspirations do not seem to be drawn from nature or naturally occurring phenomenon or systems. For instance, the Tiki-taka algorithm draws inspiration from football, which is not a naturally occurring phenomenon. Hybridizing metaheuristic algorithms is another way researchers propose new or improved algorithms. Some examples include firefly algorithm and chaos theory [32], ant colony strategy with harmony search scheme (HPSACO) [33], particle swarm optimizer combined with island-based cuckoo search, and highly disruptive polynomial mutation (iCSPM) [34], self-assembly and particle swarm optimization (SAPSO) [11], fuzzy controllers hybridized with slime mound algorithm (SMAF) [27].

3. The dwarf mongoose optimization algorithm (DMO)

The general framework of the DMO is presented, and the optimization processes are formulated.

3.1. Inspiration

The dwarf mongoose, *Helogale*, is found in areas with abundant termite mounds, rocks, and hollow trees (used for hiding), especially in Africa's semidesert and savannah bush regions. Dwarf mongoose is the smallest African carnivore with a total body length averaging 47 cm and an adult weight of approximately 400 g. They usually live

in a family group which is a matriarchy and led by an alpha pair bond for life [53,54]. In the mongoose family, the females outrank the males, and the young outrank their older siblings in each age group. The division of labor and altruism within these groups is the highest recorded for a mammal [55]. According to age and sex, different mongooses serve as guards, babysitters, attacking predators, and attacking conspecific intruders. Detailed analyses of the roles, various rank orders, and interindividual relationships have been published [56,57].

The dwarf mongooses are territorial and usually use their cheek glands and anal glands to mark upright or horizontal objects in their territory. The familiar smell of the markings makes the dwarf more self-assured and less anxious. All family members contribute to the marking of territory, and the frequency of these contributions depends on their priority rank order in the group. In a study of dwarf mongoose found in the Taru Desert, it was reported that the dwarf mongoose territory makes up areas with many termite mounds, rocks, and hollow trees. It took a family 21.8 days to cover their marked territory [58] completely. The study showed that the family size affects the coverage rate, the pattern of foraging, and sleeping mound utilization; however, the boundary marked never changed. The dwarf mongooses maintain the smallest economic area they can defend and contain sufficient resources for the group.

The dwarf mongoose has developed specific behavior and adaptations associated with territoriality and its relationship with predation avoidance. They do not have a killer bite; they instead have a skull-crush bite using the prey's eye for orientation. This killing style restricts the size of their prey, and no cooperative killing of large prey has been observed. The restrictive mode of prey capture significantly affects the mongooses' social behavior and ecological adaptations to compensate for efficient family nutrition [58]. The two central compensatory behavioral adaptations of the mongoose are as follows:

1. Prey size, space utilization, and group size

The restrictive prey catching mode of dwarf mongooses means only small prey are killed, which are not big enough for prey sharing except for feeding the young. Typical preys for the dwarf mongoose consist mostly of arthropods, sometimes small mammals, gecko lizards, and small birds are captured. The dwarf mongoose prey spectrum is a scattered food source that is unpredictable and requires an extensive search for an individual mongoose to obtain a full meal. The dwarf mongoose has adopted a seminomadic way of life that ensures the family cover relatively long distance for foraging and rarely return to a previously visited sleeping mound. This pattern of space utilization ensures that no section is over-hunted, reducing prey depletion and guarantees even predation of the entire territory [57].

The group forage as a unit with cohesion maintained through vocalization or a short nasal "peep" at 2 kHz by the alpha female [59]. The distance covered by the group each day depends on the group size, presence of young and foraging interruption caused by hiding from predators. The alpha female initiates foraging, determines the foraging path, the distance covered, and the sleeping mounds.

2. Food Provisioning

No prey-bringing to lactating females or young behavior was observed in dwarf mongooses. This has affected the social organization of the mongoose, especially parenting behavior, leading to compensatory behavioral adaptation given below.

- **Alloparenting or babysitting**

A certain number of the mongoose population (usually a mixture of males and females) serve as the babysitters. They remain with the young until the group returns at midday, where the babysitters are exchanged for the first babysitters to forage with the group. The dwarf mongooses do not build a nest for their young; they move them from one sleeping mound to another. When the young start accompanying the group, they are entirely underdeveloped, sparse pelage, and incapable of running long distances. This affects the average daily foraging thereby, restricting the group's movements.

In summary, the dwarf mongooses cannot capture large prey items, which could provide food for the whole group. The absence of a killing bite and concerted pack hunting has resulted in the dwarf mongoose adopting a social structure that enables every member to fend for itself and always move from one place to another. The dwarf mongoose has adopted a seminomadic way of life in a territory large enough to support the entire group. The nomadic behavior prevents Over-exploitation of a particular area, and it also ensures exploration of the whole territory because no previously visited sleeping mound is returned.

3.2. Population initialization

The DMO optimization begins with initializing the candidate population of the mongooses (X) as presented in Eq. (1). The population is generated stochastically between the given problem's upper bound (UB) and lower bound (LB).

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

where X is the set of current candidate populations, which are generated randomly using Eq. (2), $x_{i,j}$ denotes the position of the jth dimension of the ith population, n denotes the population size, and d is the dimension of the problem.

$$x_{i,j} = unifrnd(VarMin, VarMax, VarSize) \tag{2}$$

where *unifrnd* is a uniformly distributed random number, *VarMin and VarMax* are lower and upper bound of the problem, respectively. *VarSize* is the size of decision variables or dimensions of the problem. The best solution in each iteration is the best-obtained solution so far.

3.3. The DMO model

The proposed DMO algorithm simulates the compensatory behavioral adaptation of the dwarf mongoose. The compensatory behavior adaptation includes limitation of prey size, social organization (babysitters), seminomadic life, and many more. To implement our model, we stratify the dwarf mongoose social structure into the alpha group, scouts, and babysitters. Each group contributes to the compensatory behavioral adaptation, which leads to a seminomadic way of life in a territory large enough to support the entire group. The dwarf mongoose is known to forage and scout as a unit; for our implementation, the scouting for new mound and foraging is done simultaneously and by the same group of mongooses. As the alpha group set off foraging, they also scout for a new mound that will be visited once the babysitter exchange criterium is met. We simulate this by computing the average sleeping mound value for each iteration; depending on the value of the average sleeping mound, the next step on the dwarf mongoose population is determined as shown in Eq. (7). This nomadic behavior prevents Over-exploitation of a particular area, and it also ensures exploration of the entire territory because no previously visited sleeping mound is returned. The optimization procedures of the proposed DMO algorithm are represented in three phases, as shown in Fig. 1.

3.3.1. Alpha group

Once the population is initialized, the fitness of each solution is computed. The probability value for each population fitness is calculated by Eq. (3), and the alpha female (α) is selected based on this probability.

$$\alpha = \frac{fit_i}{\sum_{i=1}^{n} fit_i} \tag{3}$$

The number of mongooses in the alpha group corresponds to the $n - bs$. Where bs is the number of babysitters. The alpha female's vocalization that keeps the family within a path is denoted by $peep$.

The initial sleeping mound is set to $\varnothing$, and every mongoose sleep in it. To produce a candidate food position, the DMO uses the expression given in Eq. (4).

$$\boldsymbol{X_{i+1}} = X_i + phi * peep \tag{4}$$

where phi is a uniformly distributed random number [−1,1], after every iteration, the sleeping mound is as given in Eq. (5).

$$sm_i = \frac{fit_{i+1} - fit_i}{\max\{|fit_{i+1}, fit_i|\}} \tag{5}$$

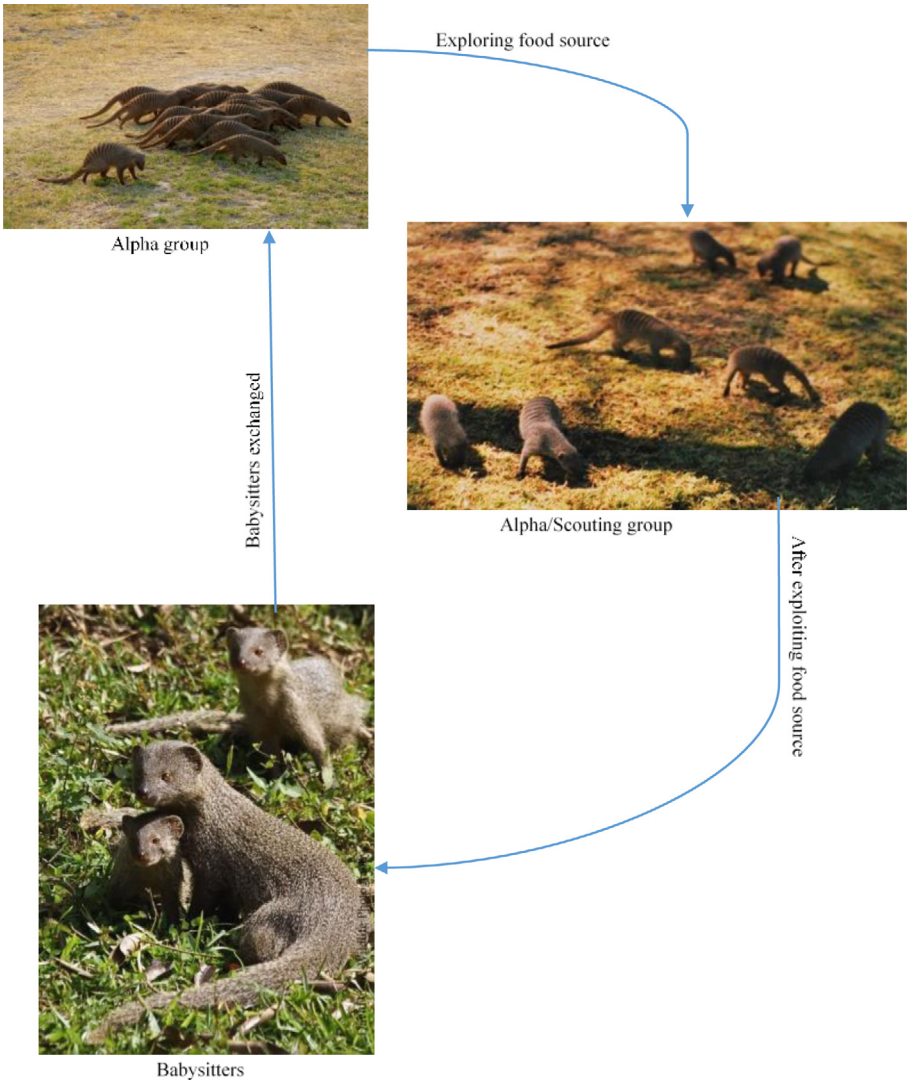

Fig. 1. The optimization procedures of the proposed DMO.

The average value of the sleeping mound found is given by Eq. (6).

$$\varphi = \frac{\sum_{i=1}^{n} sm_i}{n} \tag{6}$$

The algorithm moves to the scouting phase, where the next food source or sleeping mound is evaluated once the babysitter exchange criterium is met.

3.3.2. Scout group

The scouts look for the next sleeping mound because the mongooses are known not to return to the previous sleeping mound, which guarantees exploration. The scouting is done simultaneously with foraging for our model, as observed in [40]. This movement is modeled as an overall success or failure evaluation of finding a new sleeping mound. In other words, this movement depends on the overall performance of the mongooses. The rationale is that if the family forage far enough, they will discover a new sleeping mound. Eq. (7) simulates the scout mongoose.

$$\boldsymbol{X_{i+1}} = \begin{cases} X_i - CF * phi * rand * [X_i - \overrightarrow{\boldsymbol{M}}] & if\ \varphi_{i+1} > \varphi_i \\ X_i + CF * phi * rand * [X_i - \overrightarrow{\boldsymbol{M}}] & else \end{cases} \tag{7}$$

where,

$rand$ is a random number between [0, 1], $CF = \left(1 - \frac{iter}{Max_{iter}}\right)^{\left(2\frac{iter}{Max_{iter}}\right)}$ denotes the parameter that controls the collective-volitive movement of the mongoose group and is decreased linearly along with the iterations. $\overrightarrow{M} = \sum_{i=1}^{n} \frac{X_i \times sm_i}{X_i}$ is the vector that determines the movement of the mongoose to the new sleeping mound.

3.3.3. The babysitters

The babysitters are usually the subordinate group members that remain with the young and are rotated regularly to allow the alpha female (mother) to lead the remainder of the group on daily foraging. She usually returns at midday and in the evening to suckle the young. The number of babysitters depends on the population size; they affect the algorithm by reducing the total population size based on the percentage set. We simulate this group by reducing the population size by the percentage representative of the babysitters. The babysitter exchange parameter is used to reset the scouting and food source information previously held by the family members replacing them. The fitness weight of the babysitters is set to zero, which ensures the average weight of the alpha group in the next iteration is reduced, which means the group movement is hindered thereby, emphasizing exploitation. The pseudocode for the proposed algorithm is given in Algorithm 1.

3.4. Computational complexity

The computational complexity of the DMO typically relies on two rules: solutions initialization and the main algorithm functions consisting of the following: calculating the fitness functions, choosing the alpha female, evaluating the next sleeping mound, calculating the movement vector, and updating of solutions. Assume that n is the number of solutions, $O(n)$ is the computational complexity of the solutions' initialization processes. The computational complexity of the solutions' updating processes is $O(iter \times d \times \alpha \times sm \times \overrightarrow{M}) + O(CFE)$, which consists of exploring for the best positions, choosing the alpha female, evaluating the next sleeping mound, calculating the movement vector, and updating the solutions' positions of all solutions. The total number of iterations is called $iter$, the dimension size of the given problem is called d, and CFE is the cost of function evaluation. Accordingly, the total computational complexity of the proposed DMO is $O(iter \times d \times \alpha \times sm \times \overrightarrow{M} \times n + CFE \times n)$.

The intuitive and detailed process of DMO is shown in Fig. 2. The optimization starts with the alpha group set out for foraging, leaving the babysitters at the nest. Once a foraging site is found, they feed until midday when they return to exchange babysitters. They do not return to the previously foraged site because the scouts would have identified a new foraging site and informed the alpha female who leads the family. The babysitter exchange sets out a new exploration phase, followed by intensive exploitation till evening when the clan return to a new sleeping nest. The important point is that the DMO algorithm has only one specific parameter to be fine-tuned (the number of babysitters), and this feature is one of its superiority. To utilize the DMO algorithm, it just needs to determine the population size and the maximum number of evaluations or iterations

4. Results and discussion

We evaluate the performance of the DMO algorithm using 19 classical benchmark functions [43], 10 CEC 2020 test functions [44], and 12 benchmark problems in various fields of engineering. We then compare the results of DMO with seven different population-based metaheuristic algorithms available in the literature. The algorithms used are as follows: arithmetic optimization algorithm (AOA), hybrid constriction coefficient based (PSO) with gravitational search algorithm (GSA) referred to as CPSOGSA, Particle swarm optimization (PSO), ant colony optimization (ACO), salp swarm algorithm (SSA), sine cosine algorithm (SCA), grey wolf optimizer (GWO). The list of algorithms used and their respective control parameters are given in Table 2. We implemented the algorithms, classical and CEC 2020 test functions, and engineering design problems using MATLAB R2020b. The experiment was conducted using Windows 10 OS, Intel Core i7-7700@3.60 GHz CPU, 16G RAM. The number of independent runs for each algorithm is set at 30. Five(5) performance indicators relative to the quality of solutions obtained, namely, Best, Worst, Average, standard deviation (SD), and Median values, were used to present the results achieved by the DMO. Furthermore, statistical analysis tests were carried out using mean, standard deviation (or SD), Friedman mean ranking test, and Wilcoxon signed-rank test to make sense of the results obtained.

Algorithm 1

begin
Initialize the algorithm parameters:
[peep]
Initialize the mongoose populations (search agents): n
Initialize the number of babysitters: bs
Set n=n-bs
Set babysitter exchange parameter L
For iter=1: max_iter
Calculate the fitness of the mongoose
Set time counter C
Find the alpha based on Equation 3

$$\alpha = \frac{fit_i}{\sum_{i=1}^{n} fit_i}$$

produce a candidate food position using Equation 4

$$\boldsymbol{X_{i+1}} = X_i + phi * peep$$

Evaluate new fitness of $\boldsymbol{X_{i+1}}$
Evaluate sleeping mound using equation 5

$$sm_i = \frac{fit_{i+1} - fit_i}{\max\{|fit_{i+1}, fit_i|\}}$$

Compute the average value of the sleeping mound found using Equation 6.

$$\varphi = \frac{\sum_{i=1}^{n} sm_i}{n}$$

Compute the movement vector using

$$\overrightarrow{\boldsymbol{M}} = \sum_{i=1}^{n} \frac{X_i \times sm_i}{X_i}$$

Exchange babysitters if $C \geq L$, *and set*
Initialize bs position (Equation 1) and calculate fitness

$$fit_i \leq \alpha$$

Simulate the scout mongoose next position using Equation 7.

$$\boldsymbol{X_{i+1}} = \begin{cases} X_i - CF * rand * [X_i - \overrightarrow{\boldsymbol{M}}] & if\ \varphi_{i+1} > \varphi_i \quad Exploration \\ X_i + CF * rand * [X_i - \overrightarrow{\boldsymbol{M}}] & else \quad Exploitation \end{cases}$$

Update best solution so far

End For
Return *best solution*
End

The nature and diversity of the initial population, the population size, and the number of iterations or function evaluations can affect the performance of population-based metaheuristic algorithms [60,61]. We did a sensitivity analysis on DMO concerning the nature and size of the initial population and the number of iterations. The results showed that the performance of DMO is affected by the population size and the number of iterations. The best performance was returned when the population size was set at 50, and the maximum number of iterations was 1000. Our findings also suggested that DMO is not particularly sensitive to the different initialization schemes as presented in [61]; however, the uniform distribution recorded the best result as compared to the random number, beta, logarithm, exponential, Rayleigh, and Weibull distributions.

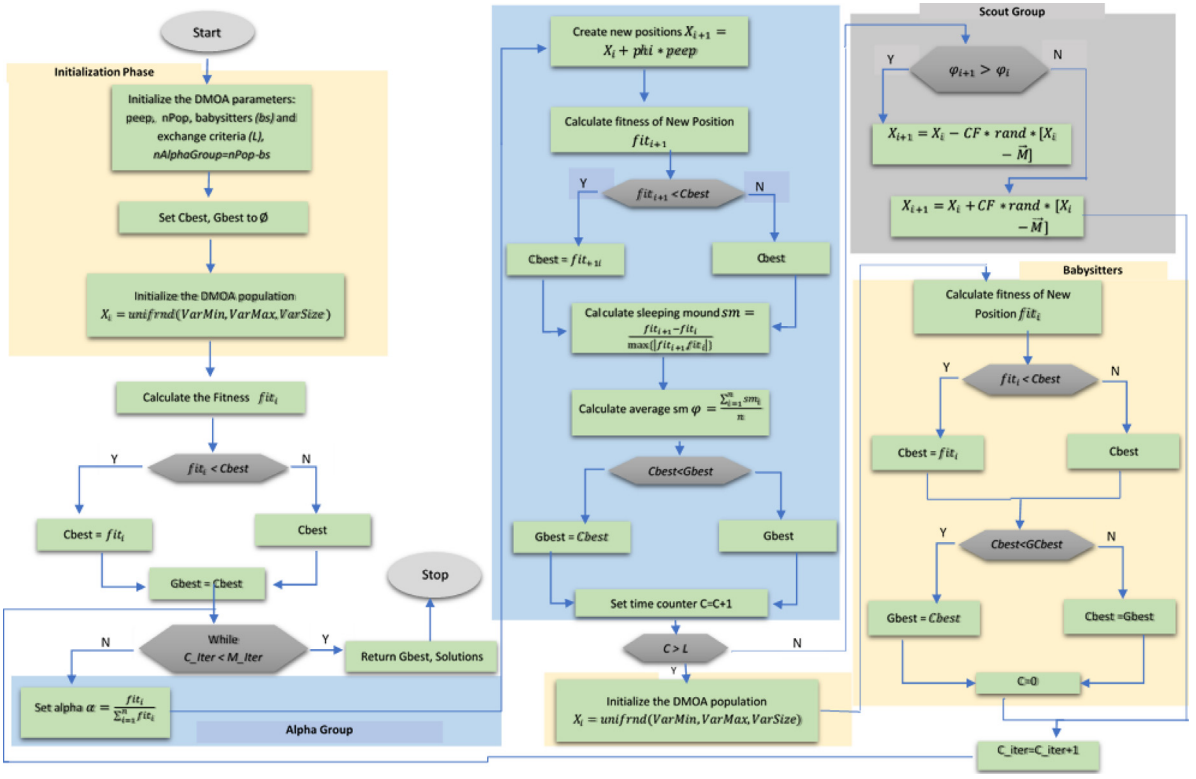

Fig. 2. Flow chart of the algorithm.

4.1. Metaheuristic algorithms

This subsection presents the state-of-the-art algorithms used to compare the proposed DMO. Seven(7) algorithms were selected based on their performance in solving related optimization problems and the availability of source code online (for replicability and scalability purposes).

Arithmetic optimization algorithm

The main inspiration of the arithmetic optimization algorithm (AOA) is the use of arithmetic operators (Multiplication, Division, Subtraction, and Addition) in solving arithmetic problems. The AOA is modeled after the rules of the arithmetic operators in mathematics. The algorithm randomly initializes the starting solutions, and the best solution for each iteration is considered the near-optimal solution [42]. The AOA has been used to solve optimization problems in many domains successfully. The control parameter of AOA used in this study is given in Table 2.

Particle swarm optimization

Particle swarm optimization (PSO) is a popular swarm intelligence algorithm that has been used with great success in different optimization domains [3]. So many variants have been proposed to solve the problem of premature convergence for some real-world complex problems. This study used the improved variants of particle swarm optimizer combined with island-based cuckoo search [30]

Constriction coefficient based PSO and GSA

The constriction based particle swarm optimization with gravitational search algorithm (CPSOGSA) is a hybrid of constriction coefficient-based particle swarm optimization and gravitational search algorithm. The goal is to combine PSO and GSA's exploitation and exploration abilities to obtain better results. The algorithm has been successfully used to solve the optimization problem in engineering [58].

Table 2
Controlling parameters.

Algorithm	Refs.	Name of the parameter	Value of the parameter
AOA	[42]	α	5
		μ	0.05
PSO	[3]	C_1, C_2 (personal and social constants)	2
		Wmax (maximum inertia weight)	0.9
		Wmin (minimum inertia weight)	0.2
CPSOGSA	[62]	$<p1$, $\langle f\rangle 2$ (control parameters)	2.05
GWO	[63]	a linearly decreased value from 2 to 0	[0,2]
		r_{1f} r_2 (random vectors)	[0,1]
SCA	[64]	a (constant)	2
SSA	[65]	c_2, c_3 (random numbers)	[0,1]
ACO	[66]	Pheromone update constant	1
		Initial pheromone	10
		Pheromone sensitivity	0.3
		Visibility sensitivity	0.1

Grey wolf optimization

The Grey wolf optimizer (GWO) is a recently proposed metaheuristic based on the social intelligence of grey wolf packs in leadership and hunting. It has been widely used for various optimization problems in many domains. The success of GWO can be attributed to its impressive characteristics over other swarm intelligence methods and the fact that it has very few parameters, and no derivation information is required in the initial search. The GWO claims simplicity, flexibility, and the right balance between exploration and exploitation during the search process [59].

Sine cosine algorithm

The Sine Cosine Algorithm (SCA) is a recently proposed optimizer for solving optimization problems. The SCA performs optimization by creating multiple initial random candidate solutions and propagating them outwards or towards the best solution using a mathematical model based on sine and cosine functions. Several random and adaptive variables integrated into this algorithm further enhance the exploration and exploitation process [60].

Salp swarm algorithm

The SSA is inspired by the swarming behavior of salps when navigating and foraging in oceans. The algorithm is modeled after the salp chain searching for optimal food sources. The chain is divided into leaders and followers. The leader initiates a chain foraging, and the followers follow it to guide them in their movements. The algorithms have been tested on several mathematical optimization functions and real-world applications successfully [61]. The SSAparameter used for this study is given in Table 2.

Ant colony optimization

The foraging behavior of ants inspires the initial design of the ant colony optimization (ACO). The ACO is modeled after ants using pheromone to mark some favorable path (food source) that other members follow. The optimization process of ACO follows a similar mechanism. The ACO was first proposed in the early nineties and is still attracting the attention of increasing numbers of researchers, and many successful applications are now available [62]. The control parameters of the respective competing algorithms used to compare the performance superiority of the DMO algorithm is presented in Table 2.

4.2. Benchmark test function

The results of each algorithm considered in this research are shown in Table 3. Nineteen (19) unimodal, fixed-dimension multimodal, and multimodal classical benchmark functions are used to test the exploitative and explorative capabilities of the DMO, as shown in the function column in Table 3. The unimodal, multimodal, and fixed-dimensional multimodal functions have different difficulties in finding the number of design variables or

Table 3
Result of classical benchmark functions.

Function	Global optima	Value	DMO	AOA	CPSOGSA	PSO	ACO	SSA	SCA	GWO
Step	0	Best	0	6.2583	0	0	9927.5	0	3.3451	7.28E−06
F1		Worst	0	6.2583	0	0	9927.5	0	4.7344	1.00E+00
		Average	0	6.2583	0	0	9927.5	0	4.0067	3.69E−01
		SD	0	0	0	0	0	0	0.34809	3.08E−01
		Median	0	6.2583	0	0	9927.5	0	4.0424	2.52E−01
Sphere	0	Best	0	0	0	0	9455	0	1.761E−08	0
F2		Worst	0	0	0	0	9455	1.365E−08	0.21316	0
		Average	0	0	0	0	9455	3.735E−09	0.0084794	0
		SD	0	0	0	0	0	5.402E−09	0.038747	0
		Median	0	0	0	0	9455	0	8.701E−05	0
SumSquares	0	Best	0	0	0	0	92 039	0	1.232E−08	0
F3		Worst	0	0	0	0	1.02E+05	1.4972	0.0023608	0
		Average	0	0	0	0	96 262	0.18353	0.000181	0
		SD	0	0	0	0	2322.6	0.33489	0.0005019	0
		Median	0	0	0	0	96 372	0.049343	1.381E−05	0
Quartic	0	Best	0	0	0	0	3.66E+07	0	0	0
F4		Worst	0	0	0	0	4.42E+07	0	4.677E−06	0
		Average	0	0	0	0	4.11E+07	0	3.926E−07	0
		SD	0	0	0	0	1.63E+06	0	9.234E−07	0
		Median	0	0	0	0	4.12E+07	0	1.545E−08	0
Matyas	0	Best	0	0	0	0	0.34	0	0	0
F5		Worst	0	0	0	0	0.34	0	0	0
		Average	0	0	0	0	0.34	0	0	0
		SD	0	0	0	0	1.69E−16	0	0	0
		Median	0	0	0	0	0.34	0	0	0
Colvile	0	Best	0	0.47742	0	0	611.1	0.0005867	0.13414	2.52E−05
F6		Worst	0	0.47742	7.8628	0	611.1	1.0161	1.6272	6.81E+00
		Average	0	0.47742	0.26284	0	611.1	0.059175	0.97986	6.84E−01
		SD	0	0	1.4354	0	1.16E−13	0.21953	0.45755	1.34E+00
		Median	0	0.47742	3.93E−06	0	611.1	0.0013499	1.0966	2.27E−03
Zakharov	0	Best	0	0	0	0	1.46E+08	0	0	0
F7		Worst	0	0	0	0	1.60E+08	0	0	0
		Average	0	0	0	0	1.53E+08	0	0	0
		SD	0	0	0	0	3.37E+06	0	0	0
		Median	0	0	0	0	1.55E+08	0	0	0
Schwefel1.2	0	Best	0	0	124.4	0	1.56E+06	0.14418	54.267	0
F8		Worst	0.0020389	0	5656.9	0	1.66E+06	138.94	7278.9	0
		Average	6.80E−05	0	1272.2	0	1.62E+06	42.493	2619.1	0
		SD	0.00037225	0	1041.3	0	23 255	32.461	2345.2	0
		Median	0	0	1075.3	0	1.62E+06	36.039	1700.9	0

(*continued on next page*)

solutions. The effectiveness and accuracy of the DMO in exploitation were tested using the unimodal, separable, and non-separable benchmark functions (F1–F9). Clearly, DMO, GOA, AOA, CPSOGSA, PSO, SSA, SCA, and GWO showed the best performance as they found the global optima for the unimodal, separable, and non-separable benchmark functions except for F9; however, ACO performed poorly. This also confirms the ability of DMO to perform exploitation effectively.

The multimodal, separable, and non-separable benchmark functions (F10–F19) were used to test the ability of the DMO to perform exploration effectively and accurately. All the algorithms considered failed to find the global optima for F14 and F17; however, DMO showed the best performance for most of the functions and was competitive in others. Friedman's test showed that DMO returned the lowest mean rank, and it ranked first in

Table 3 (*continued*).

Function	Global optima	Value	DMO	AOA	CPSOGSA	PSO	ACO	SSA	SCA	GWO
DixonPrice F9	0	Best	0.66667	0.66667	0.66667	0.24946	1.50E+08	0.66667	0.66755	0.66667
		Worst	0.66667	0.66667	0.95231	0.24946	1.68E+08	1.2786	3.395	0.66667
		Average	0.66667	0.66667	0.69643	0.24946	1.60E+08	0.79165	0.90378	0.66667
		SD	1.38E−08	0	0.071779	8.47E−17	5.43E+06	0.17945	0.5294	1.19E−06
		Median	0.66667	0.66667	0.66667	0.24946	1.59E+08	0.69417	0.69421	0.66667
HolderTable F10	−19.2085	Best	−19.209	6.72E−07	−19.209	9.9929	18.553	0	19.094	3.56E−07
		Worst	−19.209	6.72E−07	−19.209	19.17	18.553	19.186	19.209	2.22E−05
		Average	−19.209	6.72E−07	−19.209	16.997	18.553	17.82	19.205	6.50E−06
		SD	2.12E−14	0	7.94E−15	2.5661	3.61E−15	1.6914	0.02098	6.20E−06
		Median	−19.209	6.72E−07	−19.209	18.154	18.553	18.469	19.209	4.59E−06
Michalewicz5 F11	−4.687658	Best	−4.6877	2.78E+00	−4.6459	2.2993	3.4645	0	4.6877	6.57E−05
		Worst	2.6365	2.78E+00	−3.0504	4.6877	3.4645	4.6858	4.6877	1.19E+00
		Average	−2.8817	2.78E+00	−4.1473	4.1493	3.4645	3.9848	4.6877	3.03E−01
		SD	2.86E+00	0	4.15E−01	0.65132	1.36E−15	0.63786	9.034E−16	3.63E−01
		Median	−4.6877	2.78E+00	−4.3472	4.4282	3.4645	4.1497	4.6877	1.92E−01
Michalewicz10 F12	−9.66015	Best	−9.6161	5.99E+00	−9.5052	6.8839	6.8792	1.0238	9.6602	4.99E−01
		Worst	2.7506	5.99E+00	−4.3593	9.6602	7.0722	9.5303	9.6602	4.24E+00
		Average	−8.5941	5.99E+00	−7.305	8.6955	6.9503	8.1732	9.6602	1.53E+00
		SD	2.17E+00	0	1.17E+00	0.87466	5.73E−02	0.75367	1.807E−15	8.33E−01
		Median	−8.9871	5.99E+00	−7.6916	8.7521	6.9517	8.2477	9.6602	1.40E+00
Rastrigin F13	0	Best	0	0	57.708	0	9455	0.7847	7.77E−07	0
		Worst	0	0	198	0	9455	82.581	59.276	0
		Average	0	0	101.82	0	9455	52.003	14.341	0
		SD	0	0	29.695	0	0	14.934	19.074	0
		Median	0	0	98.5	0	9455	52.235	3.4398	0
Schaffer4 F14	0.292579	Best	0.02028	0.031449	0.29258	2.09E−01	0.68287	−292.28	0.0045699	0.00E+00
		Worst	0.19188	0.031449	0.29258	0.4853	0.68287	0.16351	0.19166	1.02E−07
		Average	0.1008	0.031449	0.29258	0.25919	0.68287	0.086228	0.090397	2.16E−08
		SD	0.049041	0	7.14E−17	0.082054	1.13E−16	0.04817	0.051457	2.65E−08
		Median	0.1125	0.031449	0.29258	0.21227	0.68287	0.088997	0.096973	1.59E−08
Schaffer6 F15	0	Best	0.48478	0	10.904	0	10.39	0.84353	3.3084	2.44E+00
		Worst	6.6719	0	13.289	0	11.156	9.8071	10.826	5.9285
		Average	3.3904	0	12.327	0	10.776	7.9956	8.5064	4.1315
		SD	1.6728	0	0.57845	0	0.18593	1.0794	1.6342	1.0824
		Median	3.2027	0	12.419	0	10.772	8.0537	9.0885	4.1357
DropWave F16	0	Best	0	0	−1	9.38E−01	0.80643	0	0	0
		Worst	0	0	−1	0.9494	0.80643	0.99997	0	0
		Average	0	0	−1	0.94375	0.80643	0.91815	0	0
		SD	0	0	0	0.0079818	0	0.11571	0	0
		Median	0	0	−1	0.94375	0.80643	0.91815	0	0

(*continued on next page*)

terms of overall performance for all the 19 benchmark functions. Thus, confirming the ability of DMO to solve optimization problems. Also, the values for the standard deviation for DMO are significantly smaller than the rest of the algorithms, meaning DMO is stable.

The convergence rate comparison is shown in Fig. 3, and it reveals that DMO could find the best solution among the other competing algorithms early in the iteration process. We attribute this to the attractive effect of the alpha, pulling the other members of the population towards the optimal solution in the first iteration, which is in

Table 3 (*continued*).

Function	Global optima	Value	DMO	AOA	CPSOGSA	PSO	ACO	SSA	SCA	GWO
Rosenbrock F17	0	Best	0.70132	28.147	22.406	0	4.12E+08	0.70132	28.175	2.52E+01
		Worst	24.645	28.147	90 023	0	4.14E+08	819.96	1204.4	28.509
		Average	6.1039	28.147	3059.4	0	4.13E+08	132.84	79.044	26.56
		SD	6.4371	0	16 425	0	5.64E+05	201.3	215.17	0.81019
		Median	3.5365	28.147	27.252	0	4.14E+08	29.299	29.162	26.212
Griewank F18	0	Best	0	0.092059	0.00029658	0	3.3582	0	2.015E−05	0
		Worst	0	0.092059	1.2424	0	3.3635	0.029518	0.92268	0.020259
		Average	0	0.092059	0.28118	0	3.3626	0.012637	0.17617	0.0019465
		SD	0	0	0.45161	0	0.001141	0.0082884	0.26948	0.0052828
		Median	0	0.092059	0.032588	0	3.363	0.011093	0.0073786	0
Ackley F19	0	Best	0	0	0	0	19.426	2.128E−08	0.0009	0
		Worst	0	0	19.184	0	19.426	3.1591	20.301	0
		Average	0	0	1.877	0	19.426	1.6136	15.694	0
		SD	0	0	5.4933	0	7.23E−15	0.91477	7.9872	0
		Median	0	0	0	0	19.426	1.6462	20.133	0
Chi-Square			66.406	2.97	3.47	4.26	3.39	7.47	5.05263158	6.13157895	3.24
Asymp. Sig.			0.000	1	4	5	3	8	6	7	2

the middle of the problem search area for Sphere (F2), Sum Squares (F3), and Quartic (F4). However, DMO did not converge towards the optimal solution early during the iteration process for Schwefel1.2 (F8), yet it converged to it later in the iteration process.

It is important to monitor the behavior of search agents during optimization to see how they change or improve the value of the fitness function from the initial stage to the final stage. A two-dimension implementation of six (6) functions, as shown in Fig. 4, is used to observe the behavior of search agents in DMO. The figure shows that the DMO algorithm searches around the promising regions of the search space. The distribution of the sampled points around the global optima is substantially high, which shows that the DMO algorithm exploits and explores the most promising regions of the problem search space.

4.3. The CEC 2020 test functions

This suite of functions is designed with a high difficulty level because they have many global optimum and lots of decoys. The CEC 2020 was used to evaluate the effectiveness, stability, and robustness of the DMO algorithm. All the CEC 2020 test functions were utilized for the reported experiments, and the results are presented in Table 4. The result of DMO is better than other algorithms because it was able to find the global optimum for F20, F21, F23, F25, and F27, the values for the rest are very close to the optimal values. GWO, AOA, and CPSOGSA also returned competitive results. Friedman's test result also confirmed the superiority of DMO as it ranked first with the lowest mean ranks of all the algorithms compared. The convergence rate comparison is shown in Fig. 3. DMO provides the best convergence with GWO, CPSOGSA, and AOA closely following. Here, the results are slightly worse for ACO and SCA than for classical functions. Fig. 4 shows the search history for the six benchmark functions.

4.4. Engineering problems

The newly proposed DMO algorithm was also employed to solve 12 different engineering design problems and the results are presented in this section. The engineering optimization problems used in this study find the optimal solutions under special conditions, such as design principles, resource limitations, and safety requirements. Typically, metaheuristic algorithms cannot directly find the solution to constraint optimization problems. However, equipped with constraint-handling techniques (CHTs), the optimizers can contend with the objective function and corresponding constraints. The algorithm evaluates the fitness of the candidate population using the objective function and constraints in each iteration, and the next generation of the candidate population is evaluated based on the calculated fitness function.

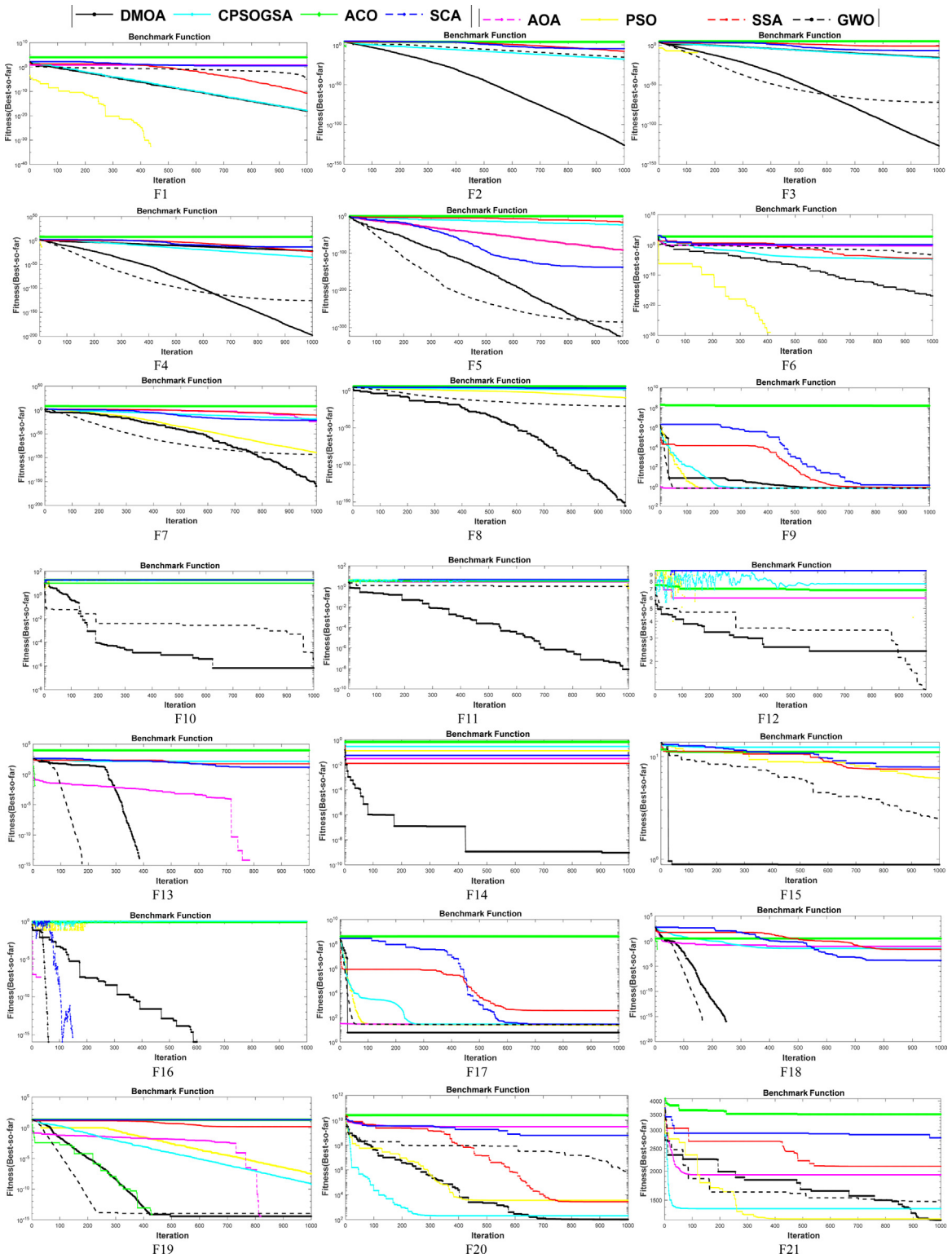

Fig. 3. Convergence rate comparison for classical and CEC 2020 benchmark functions.

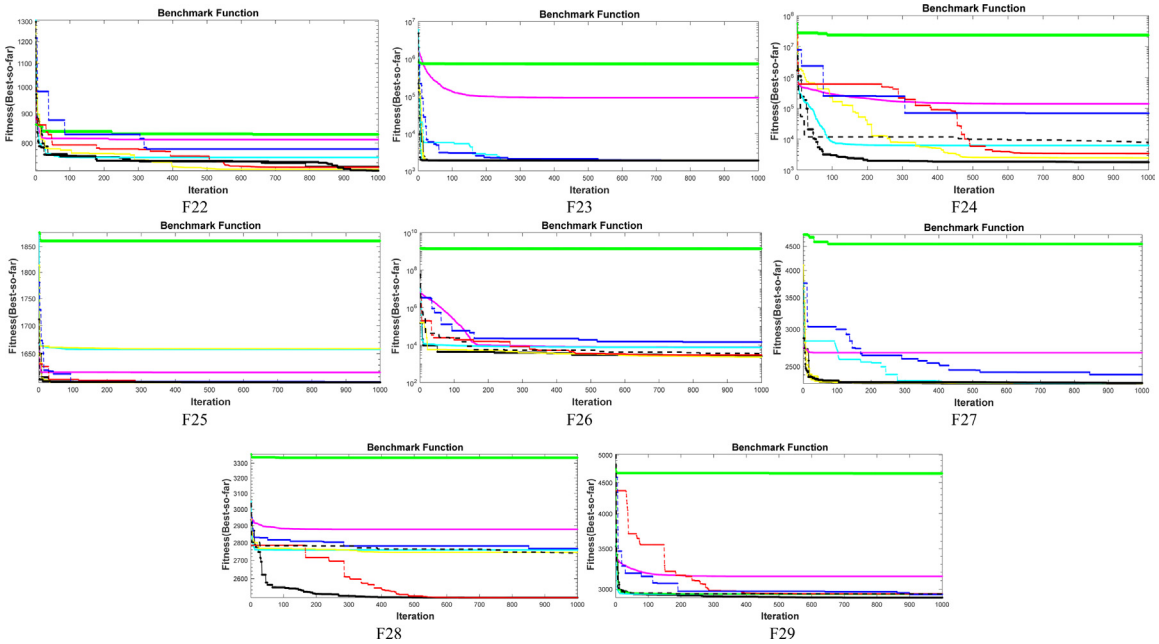

Fig. 3. (*continued*).

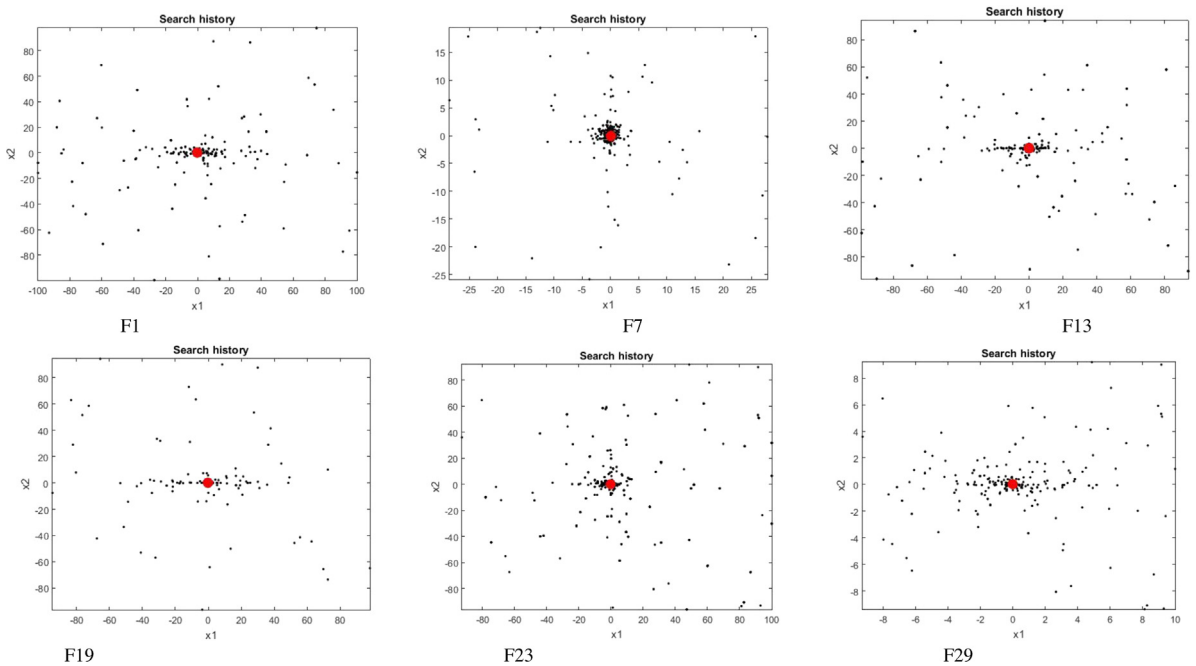

Fig. 4. Search history for six benchmark functions (F1–F6).

4.4.1. The Welded beam design problem

The Welded beam design problem is a minimization problem in which the beam is subject to vertical force [67]. The proposed DMO was used to find the minimum manufacturing cost of the design problem, and the result obtained was compared with seven other algorithms. Fig. 5 shows the schematic illustration of the WBD. The WBD is subject to the shear (τ) and beam blending (θ) stress, bar buckling load (P_c), beam end deflection (δ), and side constraints.

Table 4
Results of CEC 2020 test functions.

Function	Global Opt	Value	DMO	AOA	CPSOGSA	PSO	ACO	SSA	SCA	GWO
F20	100	Best	100.06	3.02E+09	106.97	103.04	2.57E+10	759.13	3.24E+08	5729.7
		Worst	101.8	3.02E+09	12 698	5643	2.58E+10	8959.3	1.46E+09	3.20E+08
		Average	100.43	3.02E+09	5999.7	2093.1	2.57E+10	2147.5	7.26E+08	1.62E+07
		SD	0.5024	0	4599.4	2039.7	3.13E+07	2274.4	2.59E+08	5.88E+07
		Median	100.2	3.02E+09	5508.4	1284.4	2.57E+10	1389.9	6.66E+08	27 937
F21	1100	Best	1101.4	1929.3	1378.5	1112	3335.4	1870.4	1987.3	1121.1
		Worst	1347	1929.3	2529.9	1803.4	3510	2464	2778.8	1976.7
		Average	1126.3	1929.3	2014.7	1404.5	3426.3	1859.9	2369.5	1547.3
		SD	96.47	0	283.22	176.23	43.699	322.15	178.51	204.43
		Median	1136.5	1929.3	2002.6	1441.8	3428	1801.5	2389.7	1559.4
F22	700	Best	715.3	811.84	719.81	708.63	818.4	725.3	762.47	710.94
		Worst	742.87	811.84	798.66	725.83	834.1	754.19	797.74	755.16
		Average	733.22	811.84	757.22	717.63	827.1	728.91	778.4	728.87
		SD	5.1087	0	21.584	4.3053	3.7733	12.359	8.5669	11.928
		Median	734.77	811.84	754.56	717.68	827.88	724.86	779.07	724.77
F23	1900	Best	1900.3	91 151	1900.6	1900	7.30E+05	1900.3	1907	1900.7
		Worst	1902.5	91 151	1903.3	1902.6	7.39E+05	1903.9	1969.1	1903.5
		Average	1901.2	91 151	1901.8	1901.1	7.34E+05	1901.7	1921.6	1902.3
		SD	0.54311	0	0.68664	0.52035	2416.8	0.84053	16.203	0.7511
		Median	1901.1	91 151	1901.7	1901	7.34E+05	1901.4	1915.8	1902.4
F24	1700	Best	1715.3	1.39E+05	1836.7	2032.5	2.34E+07	2755.3	10 617	2849.3
		Worst	1819.2	1.39E+05	1.73E+05	14 164	2.34E+07	17 993	69 235	3.85E+05
		Average	1749	1.39E+05	19 899	5071.5	2.34E+07	5824.5	27 377	19 155
		SD	18.67	0	36 141	3311.8	67.78	3888.4	17 830	69 228
		Median	1715.6	1.39E+05	7518	3370.7	2.34E+07	4342.9	18 894	5619.5
F25	1600	Best	1600	1617.6	1600	1600	1861.5	1600.1	1600.6	1600
		Worst	1600	1617.6	1838.8	1659	1861.5	1601.2	1602.9	1617.1
		Average	1600	1617.6	1632.7	1612.1	1861.5	1600.6	1601.1	1602.4
		SD	0.000726	0	53.109	15.15	1.16E−12	0.40604	0.45916	5.1403
		Median	1600	1617.6	1602.7	1616.8	1861.5	1600.5	1601	1600.3
F26	2100	Best	2103.8	7890.4	2589.3	2182.3	1.39E+09	2263.8	4007.4	2520.4
		Worst	2178.5	7890.4	23 934	4211.3	1.39E+09	26 256	35 695	15 961
		Average	2109.3	7890.4	8029.7	2648.2	1.39E+09	6181.6	11 483	7809.7
		SD	9.74	0	6339.6	480.71	7.443	5670.6	6484.4	4502.9
		Median	2143.9	7890.4	5047.2	2496.2	1.39E+09	3578	10 399	6824.7
F27	2200	Best	2200.1	2675.6	2242	2300	4547.2	2234.3	2276	2212
		Worst	2303.3	2675.6	2306.8	2305.3	4597.7	2307.6	3309.1	2426.5
		Average	2269.4	2675.6	2299.4	2302.2	4570.2	2300.4	2387.6	2308.5
		SD	14.228	0	12.074	1.17	12.43	14.146	177.18	29.134
		Median	2302.3	2675.6	2302.3	2302.2	4571.3	2302.6	2363.5	2306.8
F28	2400	Best	2500	2877.9	2500	2500	3335.8	2500	2542.5	2546.9
		Worst	2590.2	2877.9	2825.9	2772.3	3339.6	2778.4	2800.8	2763
		Average	2525	2877.9	2738.6	2732.1	3337.9	2740.3	2759.1	2740.7
		SD	27.203	0	96.814	44.673	1.0415	46.254	73.138	38.126
		Median	2515.2	2877.9	2769.8	2737.4	3338.1	2747.5	2781.7	2743.8

(*continued on next page*)

The design variables for WBD are: $h = x_1, l = x_2, t = x_3, b = x_4$ where the l is the length, h is the height, t is the thickness, and b is the weld thickness of the bar. The objective function of the WBD problem is mathematically formulated in Eq. (8).

$$\min f(X) = x_1^2 x_2 1.10471 + 0.04811 x_3 x_4 (14.0 + x_2) \tag{8}$$

Table 4 (*continued*).

Function	Global Opt	Value	DMO	AOA	CPSOGSA	PSO	ACO	SSA	SCA	GWO
F29	2500	Best	2598.1	3151.2	2898.3	2897.8	4649.9	2898.1	2929.4	2903.5
		Worst	2844.4	3151.2	3024.2	2946.3	4664.8	2949	2985.9	2949.7
		Average	2803.2	3151.2	2934.9	2924.5	4657.2	2923.7	2958	2938
		SD	9.2352	0	29.383	23.234	4.7329	24.085	14.63	14.205
		Median	2809.7	3151.2	2944.7	2943.5	4657.6	2943.7	2961.9	2946
Chi-Square		57.63333	1.40	6.50	4.60	2.40	8.00	3.00	5.90	4.20
Asymp. Sig.		4.47E−10	1	7	5	2	8	3	6	4

Subject to

$$
\begin{aligned}
&s_1(X) = \tau(X) - \tau_{max} \le 0,\\
&s_2(X) = \sigma(X) - \sigma_{max} \le 0,\\
&s_3(X) = \delta(X) - \delta_{max} \le 0,\\
&s_4(X) = x_1 - x_4 \le 0,\\
&s_5(X) = P - P_c(X) \le 0,\\
&s_6(X) = 0.125 - x_1 \le 0,\\
&s_7(X) = 1.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \le 0
\end{aligned}
$$

The interval for the design variables:

$$0.1 \le x_1 \le 2, \quad 0.1 \le x_2 \le 10, \quad 0.1 \le x_3 \le 10, \quad 0.1 \le x_4 \le 2$$

where

$$\tau\left(\vec{l}\right) = \sqrt{\tau'^2 + 2\tau'\tau''\left(\frac{x_2}{R}\right) + (\tau'')^2},$$

$$\tau' = P/\sqrt{2x_1x_2}, \tau'' = MR/J, M = P\left(L + \frac{x_2}{2}\right),$$

$$J = 2\left\{\sqrt{2}x_1x_2\left[\left(\frac{(x_2^2)}{4}\right) + \left(x_1 + \frac{x_3}{2}\right)^2\right]\right\},$$

$$P_c(X) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{E}/4G\right)$$

The parameters for WBD are set as follows:

$$\sigma_{max} = 3000 \text{ psi}, \text{P} = 6000 \text{ lb}, \text{L} = 14 \text{ in}, \delta_{max} = 0.25 \text{ in}, E = 3 \times 10^6 \text{ psi},$$
$$\tau_{max} = 13600 \text{ psi}, \text{ and } \text{G} = 12 \times 10^6 \text{ psi}$$

The results and statistical analysis of the eight optimizers used to solve the welded beam design problem are shown in Table 5. The DMO was able to find the global minimum cost for the WBD within the maximum iterations used for all the experiments. The SSA and GWO closely follow this; however, DMO has the lowest standard deviation that shows its effectiveness and robustness in solving this engineering problem. The convergence rate shown in Fig. 6 confirmed the superiority of the DMO in solving this problem. The Wilcoxon signed-rank test assumes that the mean distribution of the obtained results is not different. The pairwise comparison (p-values and Z scores) of DMO and all the algorithms considered returned a negative rank except that with SSA and GWO. The negative rank means that DMO has the least mean, confirming that DMO returned the best mean values and is stable around the desired result.

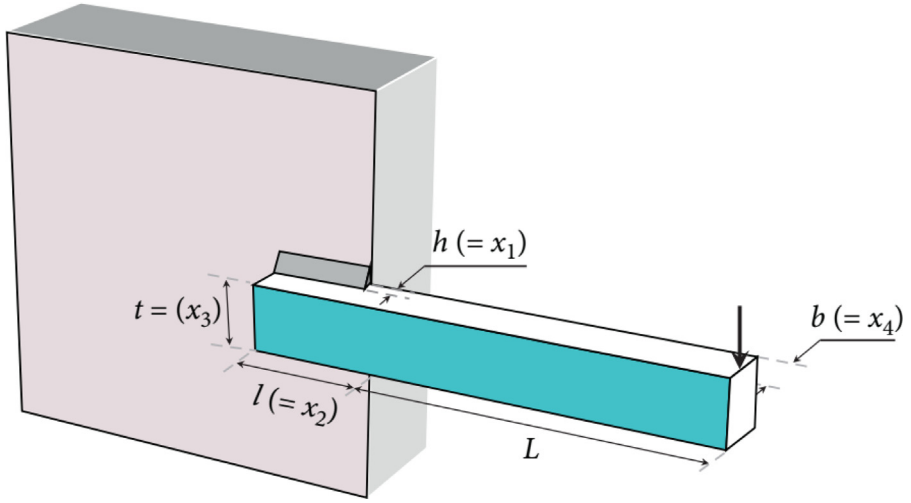

Fig. 5. Illustration of the WBD.

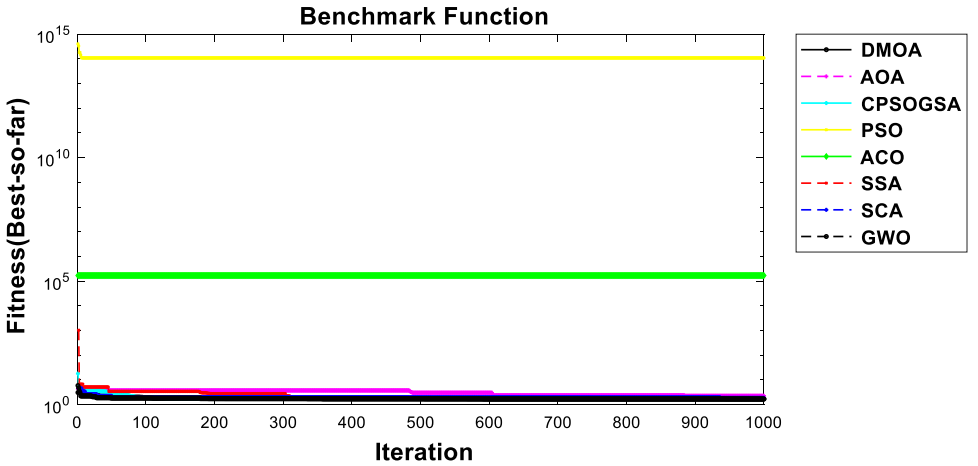

Fig. 6. Convergence rate for WBD.

Table 5
Comparative results of welded beam design.

	x1	x2	x3	x4	Best	Worst	Average	SD	Median	Z	p-values
DMO	0.2055705	3.2567724	9.036177	0.2057696	1.6953	1.7004	1.6964	0.0014007	1.6959	na	na
AOA	0.1792865	3.7660206	9.429738	0.2336683	2.1397	2.1397	2.1397	0	2.1397	−1.569[a]	0.054
CPSOGSA	0.1862275	3.6614473	8.958526	0.2093323	1.6986	2.3703	1.8998	0.18888	1.8261	−1.714[a]	0.15
PSO	0.786028	2	2	2	1.09E+14	1.09E+14	1.09E+14	0.047676	1.09E+14	−1.594[a]	0.073
ACO	1	4	3	2	1.69E+05	1.69E+05	1.69E+05	2.96E−11	1.69E+05	−1.947[a]	0.127
SSA	0.1817486	3.738515	9.036625	0.2057296	1.6953	2.0663	1.769	0.082132	1.7357	−2.019[b]	0.047
SCA	0.2074989	3.7838778	8.982884	0.2169487	1.7738	1.8776	1.8195	0.026297	1.8211	−1.859[a]	0.04
GWO	0.203239	3.302797	9.037416	0.2057535	1.6955	1.7012	1.6975	0.0015368	1.6969	−1.119[b]	0.048

[a]Based on negative ranks.
[b]Based on positive ranks.

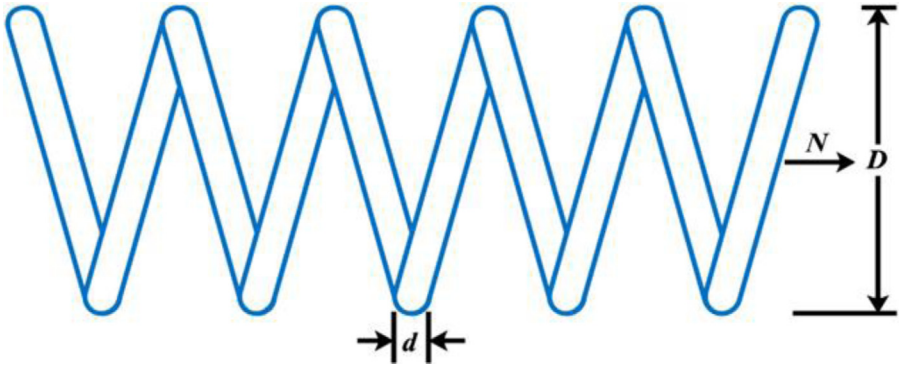

Fig. 7. Schematic illustration of the CSD problem.

4.4.2. The compression spring design problem (CSD)

The compression spring design problem (CSD) is to minimize the weight of a compression spring, as shown in Fig. 7 [68]. The problem has three design variables, namely: the number of active coils $P = x1 \in [2,15]$, the diameter of the coil or winding $D = x2 \in [0.25, 1.3]$, and the diameter of the wire $d = x3 \in [0.05, 2]$. Eq. (9) shows the model of the objective function of CSD.

Given that $l = [l_1 l_2 l_3] = [dDP]$

$$Min\ f(\overrightarrow{l}) = (l_3 + 2)\, l_2 l_1^2 \tag{9}$$

subject to

$$g_1(\overrightarrow{l}) = 1 - \frac{l_2^3 l_3}{717851^4} \leq 0,$$

$$g_2(\overrightarrow{l}) = \frac{4l_2^2 - l_1 l_2}{12566\left(l_3 l_1^3 - l_1^4\right)} + 1/5108{l_1}^2 \leq 0,$$

$$g_3(\overrightarrow{l}) = 1 - \frac{140.45 l_1}{l_2^2 l_3} \leq 0,$$

$$g_4(\overrightarrow{l}) = \frac{l_1 + l_2}{1.5} - 1 \leq 0.$$

The intervals for the design variables are:

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

The results in Table 6 compare the DMO with seven other optimization algorithms in terms of the best results and statistical analysis. The DMO, CPSOGSA, SSA, and GWO all returned the best minimum cost of the objective function. However, the DMO returned the least average and standard deviation, which implies the stability of the results of the proposed algorithm. Similarly, the convergence rate in Fig. 8 showed how the DMO, CPSOGSA, SSA, and GWO were able to find the optimum cost early in the iteration process and stabilized around the optimal cost. The Wilcoxon signed-rank test results showed that the pairwise comparison of DMO and all the algorithms considered returned a negative rank except that with SSA, CPSOGSA, and GWO. Thus confirming that DMO returned the best mean values and is stable around the desired result.

4.4.3. The pressure vessel design problem (PVD)

The pressure vessel design problem (PVD) consists of a cylindrical vessel capped at both ends by hemispherical heads. The schematic illustration of PVD is shown in Fig. 9 [69]. The thickness of the vessel T_s $(x1)$, the thickness of the head T_h $(x2)$, the vessel's inner radius $R(x3)$ and the length of the vessel barring head L $(x4)$ are the four decision variables. Eq. (10) shows the model of the objective function for PVD.

Given $l = [l_1 l_2 l_3 l_4] = [T_s T_h R L]$,

$$Min\ f\left(\vec{l}\right) = 0.6224 l_1 l_3 l_4 1.781 l_2 l_3^2 + 3.1661 l_1^2 l_4 + 19.84 l_1^2 l_3 \tag{10}$$

Table 6
Comparative results for CSD.

	x1	x2	x3	Best	Worst	Average	SD	Median	Z	p-values
DMO	0.13915	1.3	11.89243	3.6619	3.6619	3.6619	1.56E−15	3.6619	na	na
AOA	0.148288	1.3	15	6.265	6.265	6.265	0	6.265	−1.549[a]	0.128
CPSOGSA	0.13915	1.3	11.89243	3.6619	3.7303	3.6717	0.022278	3.6619	−1.591[b]	0.119
PSO	2	2	2	409.77	409.77	409.77	2.89E−13	409.77	−2.198[a]	0.029
ACO	3	2	1	209.93	209.93	209.93	5.78E−14	209.93	−1.641[a]	0.09
SSA	0.137177	1.242554	12.85631	3.6619	3.7183	3.6826	0.016854	3.6817	−1.750[b]	0.108
SCA	0.138635	1.3	11.8299	3.6635	3.7327	3.6878	0.022163	3.6793	−1.829[a]	0.063
GWO	0.139154	1.3	11.89314	3.6619	3.6619	3.6619	4.59E−06	3.6619	−1.527[b]	0.328

[a]Based on negative ranks.
[b]Based on positive ranks.

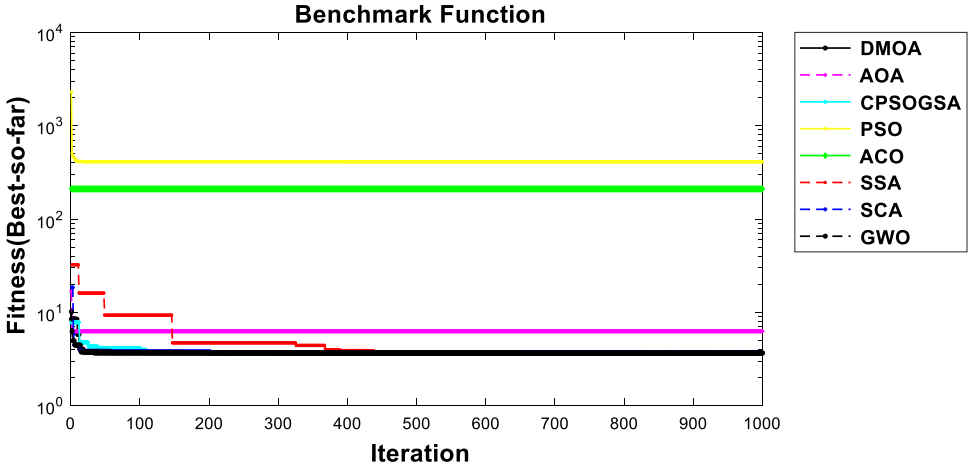

Fig. 8. Convergence rate for CSD.

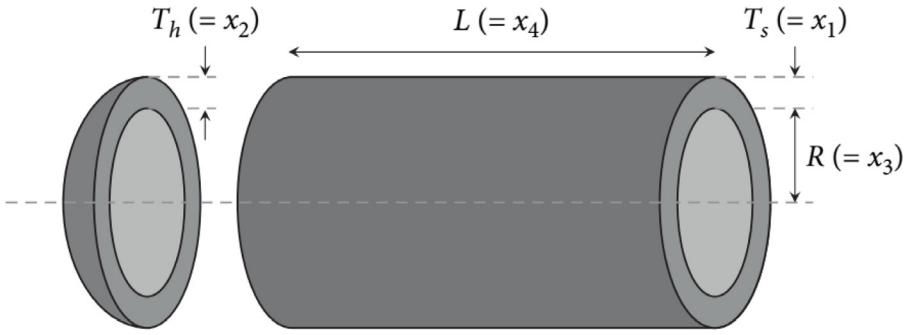

Fig. 9. Schematic illustration of the PVD.

$$g_1\left(\vec{l}\right) = -l_1 + 0.0193 l_3 \le 0,$$
$$g_2\left(\vec{l}\right) = -l_3 + 0.00954 l_3 \le 0,$$
$$g_3\left(\vec{l}\right) = -\pi l_3^2 l_4 - \frac{4}{3}\pi l_3^3 + 1296000 \le 0,$$
$$g_4\left(\vec{l}\right) = l_4 - 240 \le 0.$$

Table 7
Comparative results for PVD.

	x1	x2	x3	x4	Best	Worst	Average	SD	Median	Z	p-values
DMO	1.093571	1.04E −17	65.22523	10	2302.5	2302.5	2302.5	9.25E −13	2302.5	na	na
AOA	0.107099	0	41.30816	200	4030.5	4030.5	4030.5	0	4030.5	−1.859[a]	0.057
CPSOGSA	1.093588	0	65.22523	10	2302.5	3637.9	3153.3	580.38	3.46E+03	−2.095[b]	0.029
PSO	10	10	53.66112	71.7154	2.04E+05	2.04E+05	2.04E+ 05	1.9492	2.04E+05	−2.295[a]	0.037
ACO	1	4	3	2	1.68E+16	1.68E+16	1.68E+16	8.1368	1.68E+16	−2.173[a]	0.018
SSA	0.553754	0	43.71611	157.5719	2302.5	3638.5	3209.8	567.76	3624.6	−2.016[b]	0.041
SCA	0	0	40.32405	200	2313.8	6060.6	5063.1	1677.1	6056.8	−1.430[a]	0.191
GWO	1.090845	0.002287	65.22627	10	2302.6	6055.6	2804.3	1297.1	2302.7	−1.019[b]	0.41

[a]Based on negative ranks.
[b]Based on positive ranks.

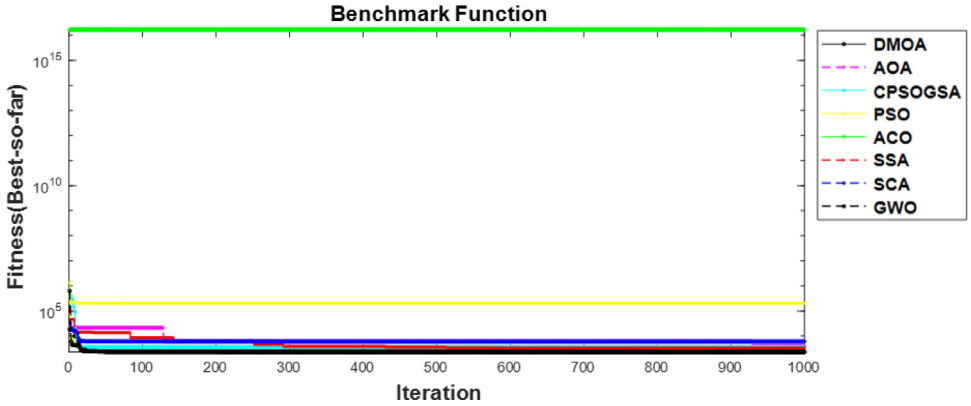

Fig. 10. Convergence rate for PVD.

The design variables defined in the interval are given below:

$$0 \le l_1 \le 99, 0 \le l_2 \le 99, 10 \le l_3 \le 200, 10 \le l_4 200$$

Similarly, Table 7 compares the DMO with seven other optimization algorithms for the best results and statistical analysis. The DMO, CPSOGSA, and SSA all returned the best minimum cost of the objective function, closely followed by GWO. Like in previous cases, DMO showed superiority by returning the least average and standard deviation. The convergence rate in Fig. 10 confirmed this superiority of DMO, closely followed by CPSOGSA, SSA, and GWO. The Wilcoxon signed-rank test results showed that the pairwise comparison of DMO and all the algorithms considered returned a negative rank except that with SSA, CPSOGSA, and GWO. This confirmed that DMO returned the best mean values and is stable around the desired result.

4.4.4. The speed reducer design problem (SRD)

The speed reducer is one of the most crucial parts of the gearbox system. The objective of this problem is to minimize the weight of the speed reducer (Fig. 11) subject to 11 constraints [70]. Seven (7) design variables characterize this design problem, namely: x_1 is the face width (b), x_2 is the module of teeth (m), x_3 stands for the number of teeth in the pinion (z), x_4 represent the length of the first shaft between bearings (l_1), (l_2) the length of the second shaft between the bearings is x_5, and the diameter of first (d_1) and second shafts (d_2) are denoted by x_6 and x_7, respectively. Eq. (11) shows the mathematical model of the objective function for this problem.

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

$$\begin{aligned} \min f(x) = {} & 0.7854 x_1 x_2^2 \left(3.3333 x_3^2 + 14.9334 x_3 - 43.0934\right) - 1.508 x_1 \left(x_6^2 + x_7^2\right) \\ & + 7.4777 \left(x_6^3 + x_7^3\right) + 0.7854 \left(x_4 x_6^2 + x_5 x_7^2\right) \end{aligned} \tag{11}$$

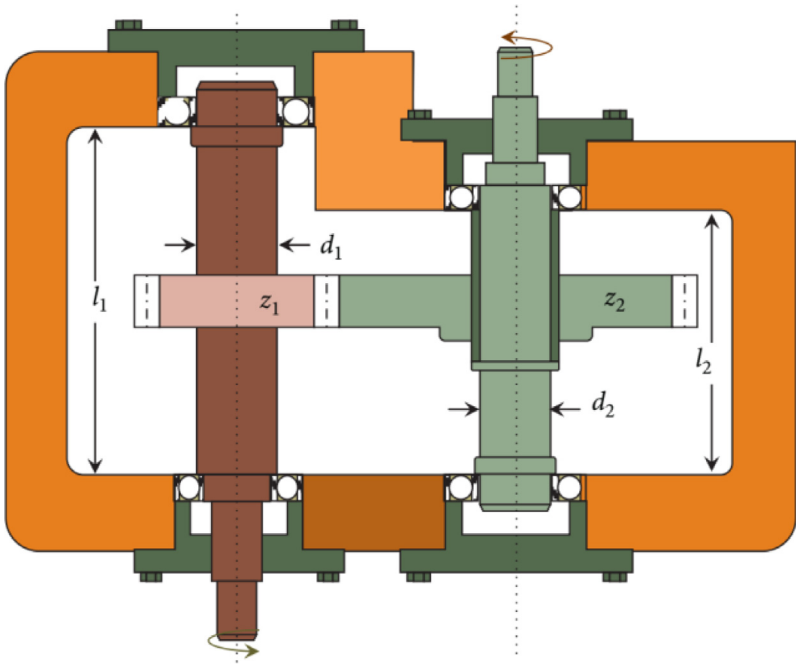

Fig. 11. Schematic illustration of the SRD.

Subject to

$$g_1(x) = \frac{27}{x_1 x_2^2 x_3} - 1 \le 0,$$
$$g_2(x) = \frac{397.5}{x_1 x_2^2 x_3^2} - 1 \le 0,$$
$$g_3(x) = \frac{1.93 x_4^2}{x_2 x_6^4 x_3} - 1 \le 0,$$
$$g_4(x) = \frac{1.93 x_5^2}{x_2 x_7^4 x_3} - 1 \le 0,$$
$$g_5(x) = \frac{\sqrt{\left(\frac{745 x_4}{x_2 x_3}\right)^2 + 16 \times 10^6}}{110 x_6^3} - 1 \le 0,$$
$$g_6(x) = \frac{\sqrt{\left(\frac{745 x_5}{x_2 x_3}\right)^2 + 157.5 \times 10^6}}{85 x_7^3} - 1 \le 0,$$
$$g_7(x) = \frac{x_2 x_3}{40} - 1 \le 0,$$
$$g_8(x) = \frac{5 x_2}{x_1} - 1 \le 0,$$
$$g_9(x) = \frac{x_1}{12 x_2} - 1 \le 0,$$
$$g_{10}(x) = \frac{1.5 x_6 + 1.9}{x_4} - 1 \le 0,$$
$$g_{11}(x) = \frac{1.1 x_7 + 1.9}{x_5} - 1 \le 0$$

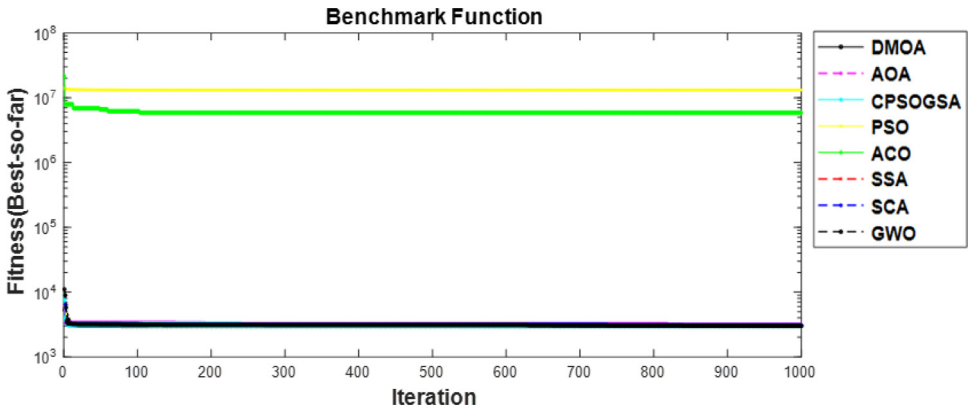

Fig. 12. Convergence rate for SRD.

The interval is given as follows:

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

The results and statistical analysis that are shown in Table 8 show that the DMO was able to find the optimal cost and returned the least average and standard deviation, which implies the stability of the results of the proposed algorithm. We noticed AOA got trapped in the local minimum. Similarly, the convergence rate in Fig. 12 showed how the DMO was able to find the optimum cost early in the iteration process and stabilized around the optimum cost. The Wilcoxon signed-rank test results returned a negative rank except that with GWO. Thus confirming that DMO returned the best mean values and is stable around the desired result.

4.4.5. The three-bar truss design problem (3-BTD)

The objective of the three-bar truss problem is to minimize the volume of loaded three-bar truss that is static subject to each bar's stress (σ) constraints. There are two design variables, namely, the cross-sectional areas, $A_1 (= x_1)\, and A_2 (= x_2)$ as illustrated in Fig. 13. The 3-BTD is mathematically modeled as Eq. (12) [71].

$$\min f(X) = \left(2\sqrt{2}x_1 + x_2\right) \times l \tag{12}$$

Subject to

$$g_1(X) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0,$$
$$g_2(X) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0,$$
$$g_3(X) = \frac{1}{\sqrt{2}x_2 + x_1} P - \sigma \leq 0$$

$l = 100$ cm. $P = 2$ kN/cm^3, $\sigma = 2$ kN/cm^3, range: $0 \leq x_1, x_2 \leq 1$

The best results and statical analysis of algorithms used are presented in Table 9. The best objective value of the DMO is equal to that of SSA and better than that of other algorithms. Except for ACO and PSO, all the tested algorithms returned values close to the global minimum cost, as shown in Fig. 14. The superiority of DMO is confirmed by the value of the average result, standard deviation, and result of the Wilcoxon signed-rank test (Z and p-value).

4.4.6. The gear train design problem (GTD)

The gear train design problem (GTD), first introduced by [72] falls under the unconstrained discrete design problem in the mechanical engineering domain. The goal is to minimize the ratio of the output/input shaft's angular

Table 8
Comparative results for SRD.

	x1	x2	x3	x4	x5	x6	x7	Best	Worst	Average	SD	Median	Z	p-values
DMO	3.497599	0.7	17	7.3	7.713535	3.350056	5.285631	2993.6	2993.6	2993.6	4.63E−13	2993.6	na	na
AOA	3.6	0.7	17	7.620008	8.3	3.520421	5.275857	3114.2	3114.2	3114.2	0	3114.2	−2.397[b]	0.019
CPSOGSA	3.497599	0.7	17	7.3	7.713535	3.350056	5.285631	2993.6	2993.6	2993.6	5.00E−13	2993.6	−1.245[a]	0.20
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	−1.256[a]	0.211
ACO	5	2	6	1	3	4	7	5.87E+06	5.87E+06	5.87E+06	2.84E−09	5.87E+06	−1.245[a]	0.20
SSA	3.497619	0.7	17	7.89341	7.925864	3.528122	5.285751	2993.6	3076.2	3017	19.782	3011.5	−2.145[a]	0.043
SCA	3.6	0.7	17	8.3	8.3	3.423716	5.338549	3029.5	3161.4	3097.1	28.334	3091	−2.711[a]	0.017
GWO	3.497891	0.7	17.00134	7.837797	7.892772	3.352131	5.286415	2994.3	3015.5	3000.7	4.0934	3000.6	−1.610[a]	0.10

[a]Based on negative ranks.
[b]Based on positive ranks.

Table 9
Comparative results for 3-BTD.

	x1	x2	Best	Worst	Average	SD	Median	Z	p-values
DMO	0.219515	0.188373	1.6953	1.7035	1.6968	0.002224	1.6958	na	na
AOA	0.216893	0.191097	2.2766	2.2766	2.2766	0	2.2766	−2.597[a]	0.026
CPSOGSA	0.261133	0.182165	1.7249	2.3461	1.8874	0.17901	1.8221	−1.285[a]	0.205
PSO	0.219515	0.188373	1.09E+14	1.09E+14	1.09E+14	0.047676	1.09E+14	−1.257[a]	0.212
ACO	1	2	1.69E+05	1.69E+05	1.69E+05	2.96E−11	1.69E+05	−1.258[a]	0.219
SSA	0.219515	0.188373	1.6953	2.0731	1.7815	0.083979	1.7576	−2.047[b]	0.040
SCA	0.21932	0.187773	1.7665	1.8948	1.8309	0.037459	1.8356	−2.731[a]	0.017
GWO	0.219506	0.188377	1.6958	1.7023	1.6974	0.001301	1.6971	−1.630[a]	0.14

[a]Based on negative ranks.
[b]Based on positive ranks.

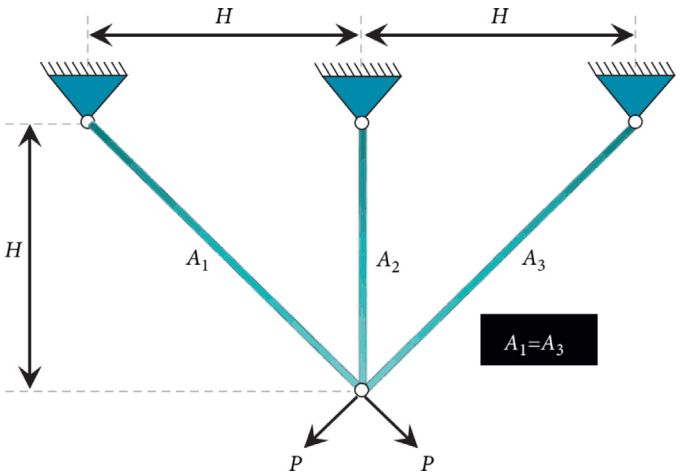

Fig. 13. Schematic illustration of the three-bar truss design problem.

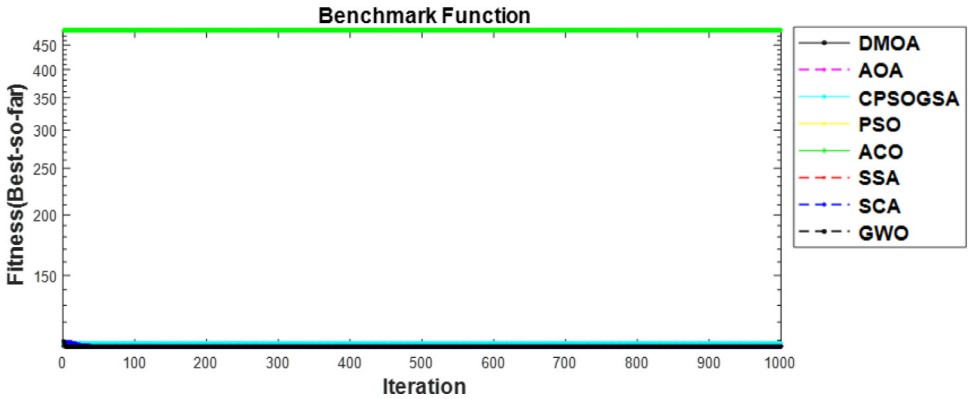

Fig. 14. Convergence rate for 3-BTD.