



Available online at www.sciencedirect.com

ScienceDirect

Comput. Methods Appl. Mech. Engrg. 415 (2023) 116200

**Computer methods
in applied
mechanics and
engineering**

www.elsevier.com/locate/cma

Mantis Search Algorithm: A novel bio-inspired algorithm for global optimization and engineering design problems

Mohamed Abdel-Basset^a, Reda Mohamed^a, Mahinda Zidan^a, Mohammed Jameel^{b,c},
Mohamed Abouhawwash^{b,d,*}

^a Faculty of Computers and Informatics, Zagazig University, Shaibet an Nakareyah, Ash Sharqia Governorate, Zagazig, 44519, Egypt

^b Department of Mathematics, Faculty of Science, Mansoura University, Mansoura 35516, Egypt

^c Department of Mathematics, Sana'a University, Sana'a 13509, Yemen

^d Department of Computational Mathematics, Science, and Engineering (CMSE), Michigan State University, East Lansing, MI, 48824, USA

Received 15 April 2023; received in revised form 10 June 2023; accepted 20 June 2023

Available online 10 July 2023

Abstract

This study presents a new nature-inspired optimization algorithm, namely the Mantis Search Algorithm (MSA), inspired by the unique hunting behavior and sexual cannibalism of praying mantises. In brief, MSA consists of three optimization stages, including the search for prey (exploration), attack prey (exploitation), and sexual cannibalism. Those operators are simulated using various mathematical models to effectively tackle optimization challenges across diverse search spaces. The performance of MSA is rigorously tested on fifty-two optimization problems and three real-world applications (five engineering design problems, and the parameter estimation problem of photovoltaic modules and fuel cells) to show its versatility and adaptability to different scenarios. To disclose the MSA's superiority, it is compared to two categories from the rival optimizers: the first category involves well-established and highly-cited optimizers, like Differential evolution; and the second category contains recently-published algorithms, like African Vultures Optimization Algorithm. This comparison is conducted using several performance metrics, the Wilcoxon rank-sum test and the Friedman mean rank to disclose the MSA's effectiveness and efficiency. The results of this comparison highlight the effectiveness of this new approach and its potential for future optimization applications. The source codes of the MSA algorithm are publicly available at <https://www.mathworks.com/matlabcentral/fileexchange/131833-mantis-search-algorithm-msa>.

Published by Elsevier B.V.

Keywords: Swarm algorithms; Global optimization; Mantis search algorithm; Constrained optimization; Unconstrained optimization

1. Introduction

Optimization is the process of finding the best possible or desirable solution(s) of a given objective function. A minimization single-objective optimization problem is defined as follows:

$$\text{Minimize } f(\vec{x})$$

* Corresponding author at: Department of Computational Mathematics, Science, and Engineering (CMSE), Michigan State University, East Lansing, MI, 48824, USA.

E-mail addresses: mohamedbasset@ieee.org (M. Abdel-Basset), redamoh@zu.edu.eg (R. Mohamed), mahinda.zidan@fue.edu.eg (M. Zidan), moh.jameel@su.edu.ye (M. Jameel), abouhaww@msu.edu, saleh1284@mans.edu.eg (M. Abouhawwash).

$$\begin{aligned} \text{Subject to: } & g_j(\vec{x}) \leq 0, \quad j = 1, 2, \dots, J \\ & h_k(\vec{x}) = 0, \quad k = 1, 2, \dots, K \\ & x_i^L \leq x_i \leq x_i^U, \quad i = 1, 2, \dots, D, \end{aligned} \tag{1}$$

where $f(\vec{x})$ is the function of the single-objective optimization problem, $\vec{x} = (x_1, x_2, \dots, x_{D-1}, x_D)^T$ is a solution that contains D decision variables ranging between the lower bound x_i^L and upper bound x_i^U , and the terms $g_j(\vec{x})$ and $h_k(\vec{x})$ are called constraint functions, in which J and K are the number of inequality and equality constraints, respectively.

Over the past few decades, several optimization techniques have been proposed to solve the optimization problem given by Eq. (1). However, with the accelerated development of human society and modern industrial and technological processes in recent years, the complexity of real-world optimization problems has increased dramatically, posing a huge challenge to optimization techniques. In general, existing optimization techniques can be divided into deterministic and meta-heuristic. Deterministic techniques are mechanical and iterative, free of randomness, and contain specified mathematical functions. This type of optimization technique requires derivative information and is therefore computationally expensive. These methods are impractical and tend to become stuck in local optima due to their complex mathematical operations. Therefore, these techniques are ineffective, particularly when handling complex problems with several peaks [1]. Newton's methods and gradient descent are classic examples of deterministic techniques.

Meta-heuristic algorithms (MAs) have emerged as a promising alternative to deterministic methods. MAs are inspired by nature, social and human behavior and use random search techniques to find the best possible solutions in the search space. However, these techniques do not guarantee that the best possible solutions will be found in a single attempt because of the stochastic search technique used by these algorithms. These techniques explore and exploit the search space by using different operators based on a minimum or maximum function [2,3]. Random search reduces the sensitivity of MAs to initial conditions and makes switching between exploitation and exploration easy. Owing to the black-box nature, we may concentrate on the input and output rather than the structural knowledge of the problems under consideration. These advantages allow MAs to identify global optimal solutions to situations that deterministic approaches cannot solve due to the lack of derivative or other relevant information.

In the past 20 years, MAs have attracted much interest from researchers for several reasons. First, they have relatively simple to execute and contain pretty straightforward concepts. Second, they perform better than local search algorithms. Third, they can be utilized in various fields, such as finance, energy, economy and trade, optimal control, image processing, scheduling, and engineering design applications [4–14]. Lastly, knowledge of the derivative of the objective function is not needed. The optimization of MAs heavily relies on exploration and exploitation [15]. MAs use exploration to seek promising positions in the search space and escape from local optima [16,17]. Meanwhile, exploiting the promising position allows these algorithms to generate a highly accurate solution [18,19]. In this case, an algorithm that combines the two elements well can avoid premature convergence in the early stages of the optimization process and can quickly converge to the global optima at the conclusion of the process.

To find the relationship between efficient optimization methods and the optimization problem they are addressing, a number of no-free lunch (NFL) theorems [20] have been developed. NFL argues that due to the diverse characteristics of these problems, any improved performance of an algorithm over one class of optimization problems does not imply that this algorithm would be effective in addressing other optimization problems with the same efficacy. As a result, several efforts have been made to create new MAs and discover those that are very strong and successful in solving different optimization problems. This circumstance prompts us to propose a novel optimization method that draws ideas from the hunting behaviors of praying mantis.

Several studies have shown that most of the proposed MAs are inspired by the behavior of animals in searching for and catching prey in nature. However, to date, no work mimics the behavior of a mantis in searching and hunting for prey. This inspired us to investigate the unique aspects of praying mantis behavior and develop mathematical models that simulate it.

In this research, we introduce a novel nature-inspired metaheuristic optimization algorithm, namely, Mantis search algorithm (MSA), which is based on the hunting and sexual cannibalism behaviors of praying mantises. The introduced MSA involves three optimization operators: search for prey (exploration), attack prey (exploitation), and sexual cannibalism to improve the exploration and exploitation operators further. It is evaluated using 52

test problems to analyze its exploration, exploitation, convergence speed, and local optima avoidance. The test results demonstrate that MSA is the most efficient compared to nine state-of-the-art optimization methods. Further validation and comparison have been made utilizing three real-world applications: five classical engineering design problems (a welded beam, a tension/compression spring, a pressure vessel, a three-bar truss, and a 10-bar truss) and the parameter estimation of photovoltaic modules and fuel cell. The results show the ability of MSA to address real-world problems with high efficiency. In sum, the main contributions of this research are listed below:

- Simulating hunting, and sexual cannibalism behaviors of praying mantises in a newly-introduced stochastic algorithm, namely the Mantis search algorithm (MSA), to tackle global optimization problems.
- Evaluating MSA using several popular benchmarks and real-world optimization problems to demonstrate its efficacy.

The shortcomings of MSA in comparison to several metaheuristic algorithms are presented in the following list:

- Having six controlling parameters that need to be accurately tuned to maximize its performance.
- Having an expensive computational cost relative to some compared algorithms

Meanwhile, it has several features that make it a strong alternative to the existing metaheuristic algorithms; these features are listed below:

- Easy to implement.
- Preserving population diversity during the optimization process.
- Having a high ability to escape from the local optima.
- Having a strong exploitation operator that enables it to solve the unimodal test functions.
- Having a high ability to balance between exploration and exploitation operators.

The rest of this article is structured as follows. Section 2 reviews the metaheuristic categories in literature. Section 3 describes the MSA introduced in this work. Test mathematical problems and optimization outcomes are discussed in Section 4. Several real-world optimization problems are presented in Section 5 together with their optimization results. Section 6 concludes the conclusions and future work directions.

2. Related work

In literature, MAs are classified into several categories. However, for simplicity, these algorithms can be classified into four main classes as follows (Fig. 1): (i) evolution-based, (ii) physics-based, (iii) swarm intelligence, and (iv) human behavior-based algorithms.

The first category of MAs comprises evolution-based algorithms (EAs) that mimic the evolutionary rules of creatures in nature. The first and most well-known EA is the genetic algorithm (GA) [21], which imitates the biological development of biological systems through mutation, recombination, and selection. The idea behind GA allowed researchers to link nature to computational skills to solve optimization problems. Since then, a large number of MAs have been created, and the list is continually expanding [19]. Several of the well-known algorithms that belong to this class are genetic programming [22], differential evolution [23], biogeography-based optimizer [24], evolution strategy [25], evolutionary programming [26], gradient evolution algorithm [27], forest optimization algorithm [28], and tree-seed algorithm [29].

The second category of MAs comprises physics-based algorithms. Physical rules, such as inertia, electromagnetic force, and gravitational force, are the source of inspiration for these algorithms. In these algorithms, the search agents communicate and move through the search space according to physical rules. Examples of these algorithms include simulated annealing [30], gravitational search algorithm [31], charged system search [32], big-bang big-crunch [33], Kepler optimization algorithm [34], black hole algorithm [35], Rime optimization algorithm [36], galaxy-based search algorithm [37], Henry gas solubility optimization algorithm [38], Ray optimization algorithm [39], central force optimization [40], artificial chemical reaction optimization algorithm [41], sine cosine algorithm [42], water cycle algorithm [43], small-world optimization algorithm [44], multi-verser optimizer [45], electromagnetism-like algorithm [46], intelligent water drops algorithm [47], ion motion algorithm [48], integrated radiation algorithm [49], river formation dynamics algorithm [50], space gravitational algorithm [51], equilibrium optimizer [52], artificial

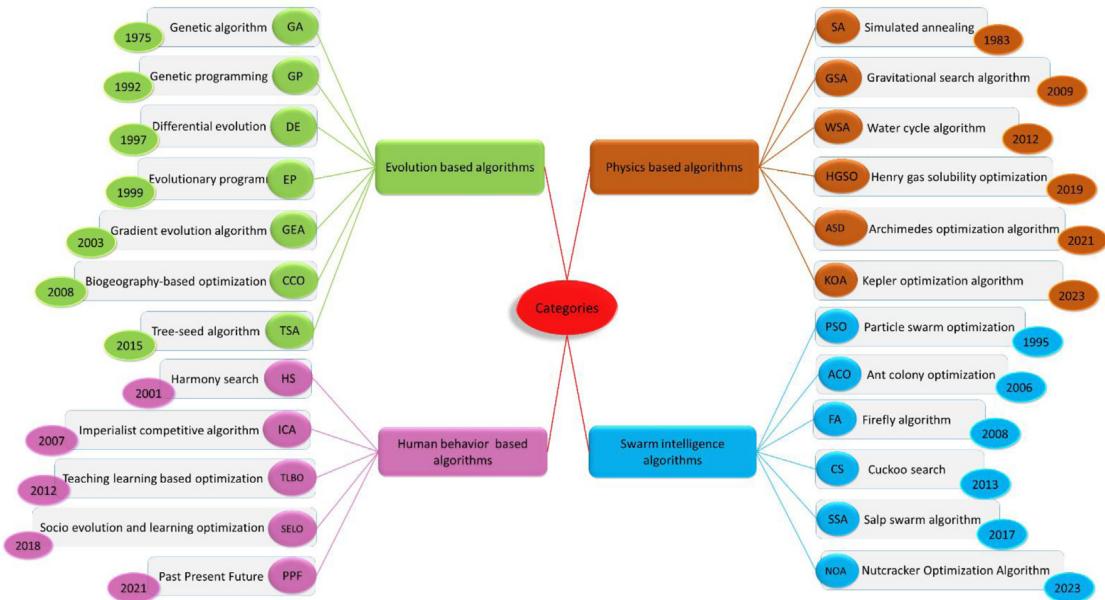


Fig. 1. Classification of MAs.

physics algorithm [53], gravitational local search algorithm [54], Archimedes optimization algorithm [55], and Young's double-slit experiment optimizer [56].

The algorithms in the third category, swarm-based, are inspired by the social behaviors of swarm intelligence (SI) in animals, birds, and insects. The most popular SI algorithm is particle swarm optimization [57]. This category includes several well-known and recently published optimization techniques, such as artificial hummingbird algorithm [1], slime mould algorithm [58], chameleon swarm algorithm [59], horse herd optimization algorithm [60], artificial gorilla troops optimizer [3], marine predators algorithm [61], nutcracker Optimization Algorithm [62], and duck swarm algorithm [63]. Other popular SI techniques are ant colony optimization [64], spider wasp optimization [65], firefly algorithm [66], bee collecting pollen algorithm [67], grey wolf optimizer [15], red fox optimization algorithm [68], whale optimization algorithm [69], bat algorithm [70], hunting search [71], salp swarm algorithm [72], cat optimization algorithm [73], krill herd algorithm [74], cuckoo search algorithm [75], donkey and smuggler optimization algorithm [76], northern goshawk optimization [77], mountain gazelle optimizer [78], elephant herding optimization [79], starling murmuration optimizer [80], and butterfly optimization algorithm [81]. A detailed review of swarm-based algorithms is presented in [82].

The last category comprises human behavior-based algorithms that are based on the social behaviors of humans. Examples include teaching–learning based optimization [83], exchange market algorithm [84], harmony search [85], political optimizer [86], past present future [87], brain storm optimization [88], and soccer league competition [89]. Several well-known human behavior-based algorithms are not listed in Fig. 1. For a detailed review of the various categories, readers can refer to [3,15,60,90].

Many methods have been recently proposed. Most of these algorithms simulate the behavior of different animals, such as foraging, mating, and hunting. In the domain of our knowledge, no study has mimicked the hunting behavior of the praying mantis in search of prey. This is our main motivation for introducing a new MA that models the behavior of the praying mantis.

3. Mantis search algorithm (MSA)

This section presents the inspiration for the proposed optimizer as well as its mathematical model and pseudo-code.

3.1. Mantises in nature

Mantises or mantodeans are among the most distinct and attractive insects. From acting as floral simulants to leaf mimics, mantises have developed various behavioral and morphological strategies to secure their prey and avoid predators. This type of insect is widespread worldwide and includes around 2400 species distributed in 434 genera [91]. This insect is characterized by its elongated body and triangular head that includes exophthalmic compound eyes and a pair of antennae, and it has a flexible neck; several species can turn their heads by 180° [92–94]. Their large eyes can spot movement as far as 30 meters away. The body length of several mantis species ranges from 10 mm to 120 mm. The complex sensorimotor behaviors of mantises have inspired many researchers to build simulation models for robot development [95,96]. Mantises have forelegs with teeth that allow them to catch and detect prey, and they have four hind legs. The forelegs are enlarged and suitable for catching and gripping prey; while standing still with the forearm flexed, the upright posture of these insects gave rise to the common name praying mantis [95,97]. Several mantises have wings. When a mantis feels threatened, it rises and spreads its wings to appear large and menacing, and in several cases, it may attempt to nip or bite the enemy [98].

Mantises generally feed on wasps, spiders, and ants. Large mantises can feed on small mantises. They can also eat small vertebrates, such as small fish, frogs, and small birds. Meanwhile, many animals, such as snakes, lizards, and small mammals, eat mantises. The front legs of mantises help trap insects, and their colors, which range from green and brown to bright colors, help them camouflage and hide among tree leaves, twigs, and flowers, thus protecting them from other predators. Many praying mantises are covert predators or ambush predators [94–96]. They can hunt by camouflage or by confrontation. The mantis uses camouflage to allow its prey to get close enough, then surprises it with a sweep attack using its highly modified front legs. While searching for food, the mantis can find prey and begin to dodge and confront it with targeted strikes using its front legs. A strange phenomenon that occurs in the mantis family is that the female, in most cases, devours the male during the mating process [99].

3.2. Mathematical model and the proposed MSA optimizer

This section presents the mathematical model of the three main stages of MSA, which are briefly described as follows. The first stage is the positioning of mantises (population initialization), which is responsible for distributing randomly the mantises within the search space of an optimization. The second stage is the exploration phase (searching for prey); this stage mimics the behaviors of mantises when they look for their prey. The third stage is the exploitation phase, which simulates the attacking behavior of mantises. The fourth stage describes sexual cannibalism, and the last stage explains the strategy of retrieving the solutions that go beyond the search space. Each of these stages is mathematically formulated in the subsequent sections.

3.2.1. Initial population

Similar to population-based algorithms, the proposed algorithm starts with the initial population of mantises. Each mantis represents a candidate solution to an optimization problem in the mantis optimization algorithm. Before starting the optimization process, a population of N mantises (solutions) in a D -dimensional search space can be expressed in a two-dimensional matrix x of size $N \times D$. The position of mantis i at function evaluation t can be defined by a vector that is randomly initialized within the lower and upper bounds of the optimization problem, as defined in the following mathematical equation:

$$\vec{x}_i^t = \vec{x}^l + \vec{r} * (\vec{x}^u - \vec{x}^l), \quad (2)$$

where \vec{x}^u and \vec{x}^l represent the upper and lower bounds of the j -dimension, respectively, and \vec{r} is a vector including numbers generated randomly between 0 and 1 according to the uniform distribution. The effectiveness of the solution is assessed for each new mantis location using a fitness function. If the solution quality of the new location is superior to that of the current one, the mantis shifts to the new location. However, if the solution quality is inferior, the MSA keeps the mantis in its current location.

3.2.2. Search for prey: exploration stage

Based on the mantis' behavior in nature, we can classify this behavior into two categories: ambush predators (sweepers) and pursuers. Ambush predators (Sweepers) usually stay motionless on tree branches or among the

weeds, and wait for prey to get to ambush distance before sweeping, whereas pursuers look for prey without using camouflage. Sweepers adopt three strategies to catch their prey: (1) concealment through camouflage, (2) quickly scanning a potentially large distance to prey, and (3) performing rapidly a surprise attack to catch prey. These strategies enable sweepers to catch a huge number of mobile prey that move within their ambush distance. Generally, the long heads with bulging eyes found in mantises play an important role in prey detection; mantises have a movable triangular head with sharp eyes that allow them to detect prey, and their eyes are located laterally, giving them a wide field of vision [100]. Mantises often use the crypsis technique to catch prey through a raptorial strike produced by their highly modified forelimbs [101]. This stage is simulated in this novel algorithm on the basis of two folds; the first one tries to simulate the behaviors of pursuers that move to other regions searching for their prey, whereas the second one seeks to mimic the behavior of speakers that wait for their prey in a concealment position before pouncing.

A. Exploration of pursuers' behavior

Starting with the pursuers, those predators search for prey away from their camouflaging positions using various steps size, long, small, and surprise orientations. This study simulates this behavior by integrating Levy flight and the normal distribution to cover small and long step sizes, as shown in Fig. 2(a), whereas surprise orientation is randomly mimicked, as shown in Fig. 2(b).

Generally, the normal distribution generates a large number, which takes the solution to a faraway position and subsequently discards numerous solutions. Meanwhile, the Levy flight generates small step sizes that consume huge function evaluations for reaching the desired solution; hence, it is not applied alone. We study hybridization between normal distribution and Levy flight to generate a different sequence of numbers that supports both small and relatively large numbers, which enables the simulation of pursuers' behaviors when searching for their prey. Fig. 2(c) presents the difference between the movements generated by the normal distribution and Levy flight and the hybridization between them. The hybridization generates steps that are not as large as those of the normal distribution and not as small as those of Levy flight (in-between them). Fig. 2(d) displays the difference between the hybrid-based steps and the pursuers' steps on the basis of integrating the random-based steps with the hybrid-based steps to simulate the surprise orientation of pursuers. The mathematical model of this behavior is as follows:

$$\vec{x}_i^{t+1} = \begin{cases} \vec{x}_i^t + \vec{\tau}_1 * (\vec{x}_i^t - \vec{x}_a^t) + |\tau_2| \cdot \vec{U} * (\vec{x}_a^t - \vec{x}_b^t), & r_1 \leq r_2 \\ \vec{x}_i^t * \vec{U} + (\vec{x}_a^t + \vec{r}_3 * (\vec{x}_b^t - \vec{x}_c^t)) * (1 - \vec{U}), & \text{Otherwise} \end{cases}, \quad (3)$$

where \vec{x}_i^t indicates the position of the i th mantis (solution) at function evaluation t , $*$ is an operator to represent the Hadamard product of two vectors, \cdot represents the multiplication operator between two numbers, $\vec{\tau}_1$ is a numerical vector generated based on the Levy flight strategy, $|\tau_2|$ is a random number based on the normal distribution with a standard deviation of 1 and a mean of 0, and r_1 , and r_2 are numbers generated randomly between 0 and 1 according to the uniform distribution. \vec{r}_3 is a vector including numerical values generated randomly at the interval $(0, 1)$ according to the uniform distribution. \vec{x}_a^t , \vec{x}_b^t , and \vec{x}_c^t are solutions selected randomly from the current population, such that $\vec{x}_a^t \neq \vec{x}_b^t \neq \vec{x}_c^t \neq \vec{x}_i^t$. \vec{U} is a binary vector generated based on the formula:

$$\vec{U} = \begin{cases} 0 & \vec{r}_4 < \vec{r}_5 \\ 1 & \text{otherwise} \end{cases}, \quad (4)$$

where \vec{r}_4 and \vec{r}_5 are vectors that include values generated randomly at the interval of 0 and 1 according to the uniform distribution; for example, each j th dimension in the vector \vec{r}_4 is compared to the same dimension in the vector \vec{r}_5 and if the former vector has a smaller value, then the j th dimension in the binary vector \vec{U} is set to 0; otherwise, it is set to 1. The first formula in Eq. (3) simulates the hybrid-based movements to explore the search space as much as possible for finding the most promising regions that might involve the near-optimal solution, and the second one is proposed to create sudden orientation based on three solutions selected randomly from the current population.

B. Exploration of speakers' behavior

The ambush mantis remains motionless and camouflaged in the trees or on the ground and waits for prey to come within striking distance. The exploration behaviors of these predators are simulated by constructing archives, including positions of a number of camouflaged places, where mantis remain within them motionless while waiting

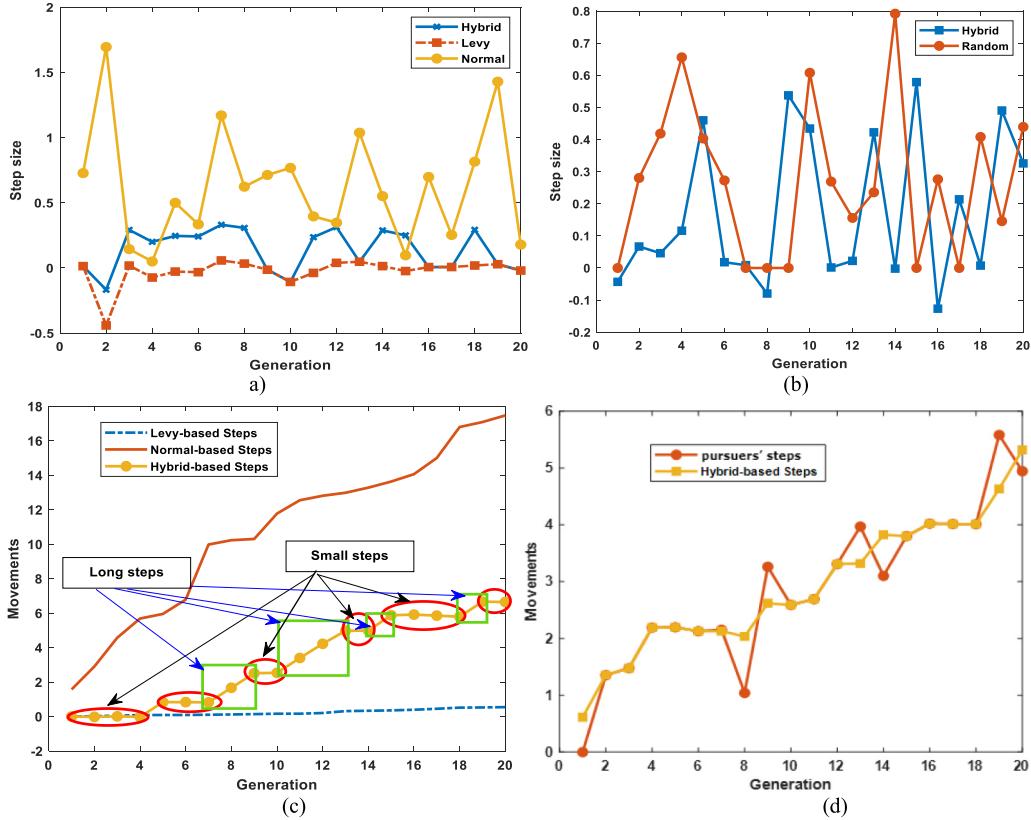


Fig. 2. (a) Comparison of Levy, normal distribution, and their hybrid. (b) Comparison of hybrid and random to achieve a random change in the mantis' exploration. (c) Comparison of Levy, normal distribution, and their hybrid in terms of movements. (d) Comparison of hybrid-based and pursuers' steps.

for the prey to come within striking distance. This archive is of size A (estimated later in the Experiments section). This archive is assigned the local best solutions of each mantis and when it is full, a solution selected randomly from it is replaced with a new one. The mantis remains motionless in their places and waits for prey to come within striking distance; it scans the surrounding environments by relying on its eyes found in the head, which can be turned by 180° . This behavior is simulated by the formula

$$\vec{x}_i^{t+1} = \vec{x}_i^t + \alpha \cdot (\vec{x}_{ar}' - \vec{x}_a^t), \quad (5)$$

where α is a factor to control the position of the mantis' head to enable the covering of the ambush distance. This factor is mathematically formulated as follows:

$$\alpha = \cos(\pi r_6) \cdot \mu, \quad (6)$$

where r_6 is a number generated randomly between 0 and 1, and μ is a distance factor and computed as follows:

$$\mu = \left(1 - \frac{t}{T}\right). \quad (7)$$

where T is the maximum number of function evaluations. The prey rapidly moves in the surrounding environments searching for food and might fall within striking distance of the mantis. Therefore, in this study, we simulate the behavior of the prey that move within the environments searching for their prey, and this behavior of getting the prey to the ambush distance is determined using the following formula:

$$\vec{x}_i^{t+1} = \vec{x}_{ar}' + (r_7 * 2 - 1) * \mu * (\vec{x}^l + \vec{r}_8 \times (\vec{x}^u - \vec{x}^l)), \quad (8)$$

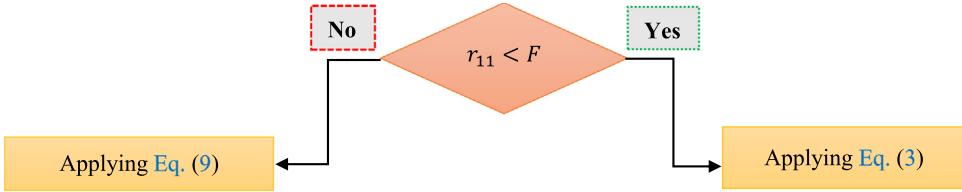


Fig. 3. Flowchart of exploration process in the proposed optimizer (r_{11} is a number generated randomly between 0 and 1).

where \vec{x}^u and \vec{x}^l represent the upper and lower bounds of the dimensions in the tackled problems, respectively; r_7 is a number generated randomly between 0 and 1; \vec{r}_8 is a vector including a number generated randomly between 0 and 1 according to the uniform distribution; and \vec{x}'_{ar} are solutions selected randomly from the archive to represent the position of the i th mantis. In Eq. (8), at the beginning of the optimization process, the distance between the camouflaged places and the prey is so high. With increasing the current iteration, this distance is gradually decreased because the prey is moved in the direction of the mantis. The ambush behaviors of mantises and their prey are mathematically formulated as follows:

$$\vec{x}_i^{t+1} = \begin{cases} \vec{x}_i^t + \alpha \cdot (\vec{x}'_{ar} - \vec{x}_i^t), & r_9 \leq r_{10} \\ \vec{x}'_{ar} + (r_7 * 2 - 1) * \mu * (\vec{x}^l + \vec{r}_8 \times (\vec{x}^u - \vec{x}^l)), & Otherwise \end{cases}, \quad (9)$$

where r_9 and r_{10} are numerical values selected randomly between 0 and 1 according to the uniform distribution to achieve a trade-off between mantises' ambush behavior and prey behavior. Then, the behaviors of pursuers and speakers are integrated within the proposed optimizer by using the recycling control factor, which divides the optimization process into parts and each one of these parts help in exploring the possible search space of an optimization problem. This factor is mathematically formulated as follows:

$$F = 1 - \frac{t\%(T/P)}{\frac{T}{P}}, \quad (10)$$

Where the symbol ‘%’ represents the remainder operator, also known as the modulus operator, and P is an integer (estimated in the Experiments section) and stands for the number of cycles, which is used to achieve a trade-off between Eqs. (3) and (9), as depicted in Fig. 3 and listed in Algorithm 1.

Algorithm 1 Exploration phase	
Output :	\vec{x}^*
1.	Initialize N mantises, \vec{x}_i^t ($i = 1, 2, \dots, N$), using Eq.(2)
2.	Evaluate each \vec{x}_i and finding the one with the best fitness in \vec{x}^*
3.	$t = 1$; //the current function evaluation
4.	while ($t < T$)
5.	r_{11} : a number created randomly between 0 and 1.
6.	Updating the recycling factor, F , using Eq. (10)
7.	for $i=1:N$
8.	if $r_{11} < F$ %% pursuers' behavior
9.	Updating \vec{x}_i^{t+1} using Eq. (3)
10.	Else %% Speakers' behavior
11.	Updating \vec{x}_i^{t+1} using Eq. (9)
12.	End if
13.	$t = t + 1$
14.	Evaluate the mantis, \vec{x}_i^{t+1} , and replace \vec{x}_i^t with, \vec{x}_i^{t+1} if it is better.
15.	End for
16.	End while

3.2.3. Attacking the prey: Exploitation stage

Mantises are stealthy and ambush predators, so the movement of their front legs and the quick capture of prey are crucial for survival. They are fast enough to catch mosquitoes and flies in mid-air. The main inspiration of the mantis search algorithm originates from hunting behavior, which consists of attacking the prey and capturing it. The behavior of the mantis in catching prey can be classified into two stages, namely, the approach and the

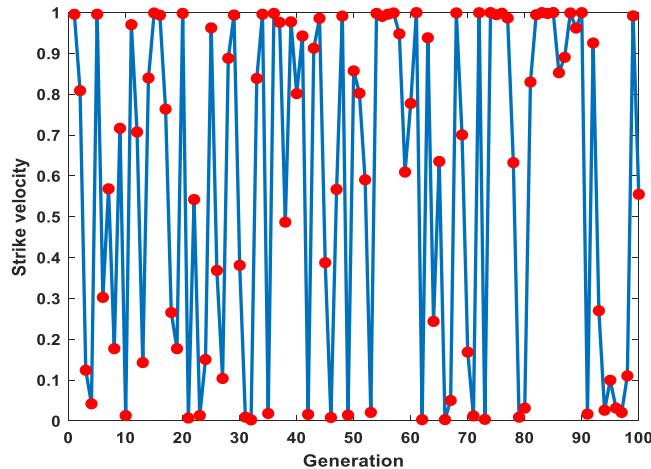


Fig. 4. Motion of the mantis during the attack on the prey.

sweep [102–105]. In the approach stage, a mantis extends its arms up and outward. In the sweep stage, the mantis catches the prey at high speed and then eats it.

A mantis can estimate the distance between itself and its prey before deciding to sweep (strike) [106–109]. Several studies have shown a linear relationship between the average strike distance and the length of the foreleg of a predator [107,108,110]. In mantis literature, prey distances are commonly reported not in centimeters but as a percentage of the length of the forelegs. These studies have shown that the maximum motivation distance to strike is between 70% and 80% of the foreleg length. Effective striking is accomplished at a distance of 30%–40% of the foreleg extension. This species of insect can estimate the distance to its prey through a precise triangulation mechanism [106,107,111–114]. As mentioned previously, the mantis has a triangular head and two large compound eyes. Within the restricted range of distances within which a mantis can catch prey with its raptorial forelegs, the acute zone and the comparatively large binocular field of the compound eyes allow distance assessment using binocular triangulation.

Several studies have shown that coordinated movements during orienting are only ballistic events; these are pre-determined based on visual and proprioceptive data before the onset of instructing [106,107,111,115]. The mantis rises (before the strike) at a suitable angle and attacks the prey at high velocity. If a mantis initially miscalculates the speed of its prey, the mantis will often correct its mistake with a similar pause. Thus, the estimation of the distance between predator and prey (strike distance) and the velocity of the attack (strike velocity) represent two fundamental pillars in the success of the hunting process. In this study, the hunting behavior of the mantis in nature and its efficient methods of capturing prey are simulated by building mathematical models developed to design the introduced algorithm. The mathematical model of this stage is discussed in the rest of this section.

Once the mantis perceives its prey, it begins to use its excellent methods of catching and consuming it. These methods consist of estimating the distance of the strike and the velocity of the prey's strike. To mathematically simulate this behavior, the following steps are suggested.

- Estimating the strike distance (d_s).
- Calculating the strike velocity (v_s).
- Considering the strike's failure; in this case, the mantis attempts to strike again.

A. Striking velocity

The mantis finishes hunting by attacking its prey when it is too close to it. The mantis is believed to have the capability to know when it is time to invade its prey. The mantis uses its front legs to attack its prey. Thus, its position is updated by anchoring its hind legs and extending its front legs as far as possible toward the prey. This mechanism helps the mantis to exploit the search space and capture its prey. We use the sigmoid function with a constant value to calculate the mantis strike velocity magnitude when attacking prey, as shown in Fig. 4. The

striking velocity magnitude of the front legs of a mantis toward its prey can be mathematically modeled with the following equation:

$$v_s = \frac{1}{1 + e^{l\rho}}, \quad (11)$$

where v_s represents the strike velocity of the mantis and ρ represents the gravitational acceleration rate of the mantis's strike, which is a constant value determined in the subsequent experiments. l is a number generated between -1 and -2 to control the gravitational acceleration rate; any value of l approaching -2 and -1 maximizes and minimizes the striking velocity magnitude v_s close to 1 and close to 0, respectively. On one hand, when v_s approaches to 0, the mantis sees that it is not a suitable time to attack the prey; On the other hand, approaching a value of 1 makes the mantis move quickly to attack the prey to be captured and consumes it before its escape. Each mantis' behavior of catching the prey is updated by the following formula:

$$x_{i,j}^{t+1} = (x_{i,j}^t + x_j^*) / 2.0 + v_s \cdot d_{si,j}^t, \quad (12)$$

where $x_{i,j}^{t+1}$ represents the new position for the j th dimension of mantis i at function evaluation $t + 1$, v_s determines the striking velocity magnitude of the mantis which is re-positioned between its current position, $x_{i,j}^t$ the prey's position to reduce the distance between them as much as possible and to facilitate the attacking process, and $d_{si,j}^t$ represents the strike distance. The strike distance varies in accordance with the size of the praying mantis. In other words, the larger the mantis is, the greater the strike distance is, and vice versa. In the proposed algorithm, $d_{si,j}^t$ is computed according to the following formula:

$$d_{si,j}^t = (x_j^* - x_{i,j}^t), \quad (13)$$

where x_j^* indicates the position for the j th dimension of the prey or the best solution obtained by far and \vec{x}_i^t represent the current position for the j th dimension of the i th mantis. In this equation, the strike distance is great when $x_{i,j}^t$ is far from the prey; otherwise, the strike distance is small.

Sometimes, the strike of the mantis fails, and it needs to change its direction before striking again. This behavior is formulated as the following formula, where the mantis changes its direction based on the direction of two mantises selected randomly from the population.

$$x_{i,j}^{t+1} = x_{i,j}^t + r_{12} \cdot (x_{a,j}^t - x_{b,j}^t), \quad (14)$$

where x_a^t and x_b^t are two mantises selected randomly from the current population to determine the direction of the current mantis before it strikes again and r_{12} is a number selected randomly between 0 and 1 according to the uniform distribution. Failure of the mantis strike means the mantis has fallen into the trap of the local optima. Thus, the individuals trapped in the local optima need a significant exploration and exploitation capability to escape from the optima. The following mathematical model is proposed to prevent the algorithm from falling into the local optima by updating the mantises to take better positions for striking the prey again.

$$x_{i,j}^{t+1} = x_{i,j}^t + e^{2l} \cdot \cos(2l\pi) \cdot |x_{i,j}^t - \vec{x}_{ar,j}^t| + (r_{13} \cdot 2 - 1) \cdot (x_j^u - x_j^l) \quad (15)$$

where r_{13} is a numerical value generated randomly between 0 and 1 according to the uniform distribution. In the proposed algorithm, Eq. (15) is used with a failure probability in order to avoid getting trapped in local minima and to enhance the convergence rate towards the optimal solution for an optimization problem. The probability formulated in Eq. (16) gradually decreases with increasing current function evaluation to decrease the exploration process, which is maximized at the outset of the optimization process, and increase gradually the exploitation operator to accelerate the convergence toward the near-optimal solution.

$$P_f = a \cdot (1 - \frac{t}{T}), \quad (16)$$

where a is a predefined fixed value ranging between 0 and 1 that controls exploration and exploitation operators. A high value for this parameter increases the exploration and minimizes the exploitation and vice versa. Eqs. (12) and (14) are randomly exchanged within MSA, and Eq. (15) is applied after Eq. (12) according to the probability discussed before in Eq. (12). Fig. 5 and Algorithm 2 show how the three equations are arranged inside the proposed optimizer to balance the exploitation and exploration capabilities in addition to overcoming the local minima problem. In Fig. 5, r_2 is a random number in the interval $[0, 1]$ for each solution, and it is constant for all dimensions in each solution; r_4 is a number generated randomly for each dimension in the updated solution.

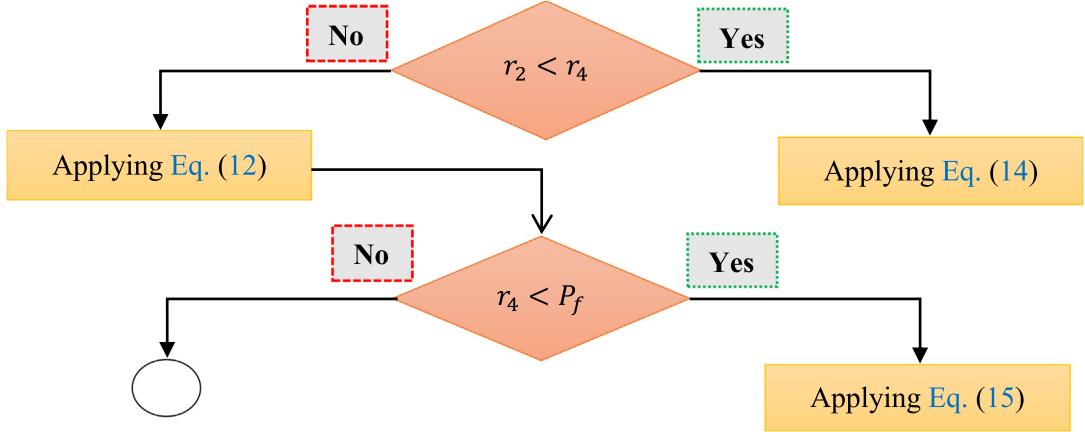


Fig. 5. Flowchart of attacking prey (exploitation) process in the proposed optimizer.

Algorithm 2 Exploitation phase

Output : \vec{x}^*

1. Initialize N mantises, \vec{x}_i^t ($i = 1, 2, \dots, N$), using Eq. (2)
 2. Evaluate each \vec{x}_i and finding the one with the best fitness in \vec{x}^*
 3. $t = 1$; //the current function evaluation
 4. **while** ($t < T$)
 5. r_4 : a number created randomly between 0 and 1.
 6. **for** $i=1:N$
 7. **for** $j=1:D$
 8. r_2 : a number created randomly between 0 and 1.
 9. **if** $r_2 < r_4$
 10. Updating x_{ij}^{t+1} using Eq. (14)
 11. **Else**
 12. Updating x_{ij}^{t+1} using Eq. (12)
 13. **if** $r_4 < P_f$
 14. Updating x_{ij}^{t+1} using Eq. (15)
 15. **End**
 16. **End if**
 17. **End for**
 18. $t = t + 1$
 19. Evaluate the mantis, \vec{x}_i^{t+1} , and replace \vec{x}_i^t with, \vec{x}_i^{t+1} if it is better.
 20. Updating the failure's probability P_f using Eq. (16)
 21. **End for**
 22. **End while**
-

3.2.4. Sexual cannibalism

In praying mantises, the females eat the males during or after copulation; this behavior is known as sexual cannibalism. The first operation in this behavior is mate attraction by females of praying mantis to attract the males to their locations [116], which is mathematically simulated in accordance with the formula

$$\vec{x}_i^{t+1} = \vec{x}_i^t + \vec{r}_{16} * (\vec{x}_i^t - \vec{x}_a^t), \quad (17)$$

where \vec{x}_i^t represents the female of praying mantises, \vec{x}_a^t is a solution selected randomly from the population to represent the male attracted by the female for mating and then being eaten, and \vec{r}_{16} is a vector including random numbers in the interval $[0, 1]$ according to the uniform distribution to stand for the attraction factor, where the mated females attract the males at a low rate contradictory to the virgin females that attract males with a high probability. Mate attraction is performed at the outset of the optimization process to attract the male to the female's location, so it is performed with a high probability at the beginning of the optimization process; this probability decreases

gradually when the current generation is increased. Generally, the mathematical formulation of this probability, namely P_t , is as follows:

$$P_t = r_{17} \cdot \mu. \quad (18)$$

where r_{17} is a random number generated between 0 and 1 according to the uniform distribution. Afterward, the male mates with the female via the uniform crossover operator borrowed from the genetic operators to produce a new offspring, as described in the following formula:

$$\vec{x}_i^{t+1} = \vec{x}_i^t * \bar{U} + (x_{11}^t + \vec{r}_{18} * (-x_{11}^t + \vec{x}_i^t)) * (1 - \bar{U}), \quad (19)$$

where \vec{r}_{18} is a vector including random values generated between 0 and 1 according to the uniform distribution. x_{11}^t represents the value of the 1th dimension in the 1th mantis and this value represents the male that mates with the female. We make this stationary to cover the dimensions that load the converged values within different dimensions. After or during mating, the female eats the male using the following formula:

$$\vec{x}_i^{t+1} = \vec{x}_a^t \cdot \cos(2\pi l) \cdot \mu, \quad (20)$$

where \vec{x}_a^t represents the male, μ is the eaten part from the male, and $\cos(2\pi l)$ is used to give the female flexibility to turn the male around during the eating process. The male behavior focuses on reducing the risk of attack caused by the female, and several males reduce the risk of being cannibalized. Therefore, the cannibalism of males by females is applied with a probability generated randomly between 0 and 1, as found in Algorithm 3, which describes the steps of sexual cannibalism.

Algorithm 3 Sexual cannibalism

1. Initialize N mantises, \vec{x}_i^t ($i = 1, 2, \dots, N$), using Eq. (2)
2. Evaluate each \vec{x}_i and finding the one with the best fitness in \vec{x}^*
3. $t = 1$; //the current function evaluation
4. **while** ($t < T$)
 5. **for** $i=1:N$
 6. r_3 : a number created randomly between 0 and 1.
 7. r_4 : a number created randomly between 0 and 1.
 8. **if** $r_3 < r_4$ % % Mating behavior
 9. Updating \vec{x}_i^{t+1} using Eq. (19)
 10. **Else**
 11. **if** $r_4 < P_t$ % % Mate attraction
 12. Updating \vec{x}_i^{t+1} using Eq. (17)
 13. **Else** % % Cannibalism process
 14. Updating \vec{x}_i^{t+1} using Eq. (20)
 15. **End**
 16. **End if**
 17. $t = t + 1$
 18. Evaluate the mantis, \vec{x}_i^{t+1} , and replace \vec{x}_i^t with, \vec{x}_i^{t+1} if it is better.
 19. Updating P_t using Eq. (18)
 20. **End for**
 21. **End while**

3.2.5. Proposed algorithm: MSA

This section describes the introduced algorithm in detail. At the outset and before starting the optimization process, all required parameters are inputted to the algorithms; these parameters are T (maximum iteration), N (population size), A (length of an archive), a (probability of the strike's failure), p (probability to exchange between the exploration and exploitation stages as determined by the sensitivity analysis according to the optimization problem), P (recycling factor to exchange between pursuers and speakers), ρ (gravitational acceleration rate of the mantis's strike), and P_c (used to determine the percentage of sexual cannibalism). All these parameters, except N and T , are analyzed in the next section to estimate the best value for maximizing the performance of the introduced algorithm.

Algorithm 4 The steps of the proposed MSA

Input: T, N, A, a, P_c , ρ , p , and P

Output : \vec{x}^* , the best mantis and its fitness value

1. Initialize N mantises, \vec{x}_i^t ($i = 1, 2, \dots, N$), using Eq. (2)
2. Evaluate each \vec{x}_i and finding the one with the best fitness in \vec{x}^*
3. $t = 1$; //the current function evaluation
4. **while** ($t < T$)
5. r: a number created randomly between 0 and 1.
6. **if** $r < p$ %%% Exploration phase
7. r_1 : a number created randomly between 0 and 1.
8. Updating the recycling factor, F , using Eq. (10)
9. **for** $i=1:N$
10. **if** $r_1 < F$ %% Pursuers' behavior
11. Updating \vec{x}_i^{t+1} using Eq. (3)
12. **Else** %% Spearers' behavior
13. Updating \vec{x}_i^{t+1} using Eq. (9)
14. **End if**
15. $t = t + 1$
16. Evaluate the mantis, \vec{x}_i^{t+1} , and replace \vec{x}_i^t with, \vec{x}_i^{t+1} if it is better.
17. **End for**
18. **Else** %% Exploitation phase
19. r_4 : a number created randomly between 0 and 1.
20. **for** $i=1:N$
21. **for** $j=1:D$
22. r_2 : a number created randomly between 0 and 1.
23. **if** $r_2 < r_4$
24. Updating x_{ij}^{t+1} using Eq. (14)
25. **Else**
26. Updating x_{ij}^{t+1} using Eq. (12)
27. **if** $r_4 < P_f$
28. Updating x_{ij}^{t+1} using Eq. (15)
29. **End**
30. **End if**
31. **End for**
32. $t = t + 1$
33. Evaluate the mantis, \vec{x}_i^{t+1} , and replace \vec{x}_i^t with, \vec{x}_i^{t+1} if it is better.
34. Updating the failure's probability P_f using Eq. (16)
35. **End for**
36. **End If**
37. **if** $r < P_c$ %%% %% Sexual cannibalism
38. **for** $i=1:N$
39. r_3 : a number created randomly between 0 and 1.
40. r_4 : a number created randomly between 0 and 1.
41. **if** $r_3 < r_4$ %% Mating behavior
42. Updating x_i^{t+1} using Eq. (19)
43. **Else**
44. **if** $r_4 < P_t$ %% Mate attraction
45. Updating x_i^{t+1} using Eq. (17)
46. **Else** %% Cannibalism process
47. Updating x_{ij}^{t+1} using Eq. (20)
48. **End**
49. **End if**
50. $t = t + 1$
51. Evaluate the mantis, \vec{x}_i^{t+1} , and replace \vec{x}_i^t with, \vec{x}_i^{t+1} if it is better.
52. Updating P_t using Eq. (18)
53. **End for**
54. **End if**
55. **End while**

Afterward, a population consisting of N mantises is distributed within the search space of the problem. The fitness value of each solution is computed, and the fittest one is assigned to \vec{x}^* to represent the best solution so far. Then, the optimization process that involves three main phases of exploration, exploitation, and sexual cannibalism is implemented to update the initialized solution for obtaining improved outcomes for the optimization problems. The optimization process is repeated until satisfying the termination conditions. The shortcoming of this optimizer is that it has six controlling parameters (A , a , P_c , ρ , p , and P) that need to be accurately tuned to maximize its performance. Meanwhile, it has several advantages that make it a strong alternative to the existing metaheuristic algorithms, some of these advantages are listed below:

- Easy to implement.
- Preserving population diversity during the optimization process.
- Having a high ability to escape from the local optima.
- Having a strong exploitation operator that enables it to solve the unimodal test functions.
- Having a high ability to balance between exploration and exploitation operators.

The mantis optimizer strikes a balance between exploration and exploitation by selecting either search for prey or attacking the prey based on a predefined probability value (p), estimated later in the sensitivity analysis section. The time complexity of MSA is $O(NDT)$, where N , D , and T are the main factors that affect the time complexity of the suggested optimizer. In conclusion, the pseudocode and flowchart of MSA are presented in Algorithm 4 and Fig. 6 respectively.

4. Results and discussion

This section first observes the performance of the proposed algorithm on 52 well-known mathematical test functions to show its effectiveness compared to two categories from rival optimizers: the first category contains the well-established algorithms such as whale optimization algorithm (WOA) [69], differential evolution (DE), and Grey wolf optimizer (GWO, 2014) [15]; and the second category includes some of the recently-published algorithms, such as Equilibrium optimizer (EO, 2020) [52], African vultures optimization algorithm (AVOA, 2021) [117], Artificial gorilla troops optimizer (GTO, 2021) [3], Gradient-based optimizer (GBO, 2020) [118], weighted mean of vectors (INFO, 2022) [119], and Runge Kutta method (RUN, 2021) beyond the metaphor [120]. The parameters of those algorithms according to their original references are described in Table 1. These test functions are divided into three categories: six unimodal (F1–F6), seven multimodal (F7–F13), and ten fixed-dimension multimodal (F14–F23), which are described in terms of the equations, number of dimensions, search space, and topology in [62]. Broadly speaking, the unimodal test functions include only a global best position, thus they can assess the exploitative behavior of the optimization algorithms. Meanwhile, the multimodal test functions involve several local optima that have to be avoided by the optimization algorithms to reach the global one; therefore these functions can be employed to observe the explorative behavior and local optima avoidance capabilities of these algorithms.

All the algorithms used in our experiments underwent 30 independent runs, with each run continuing until 50,000 function evaluations. The results from these runs were analyzed to calculate five performance metrics: the best, average, and worst fitness values, as well as the standard deviation (SD) and Friedman rank sum (F-rank). Furthermore, both the Wilcoxon rank-sum test and convergence curve metrics were used to demonstrate the significant difference in performance between the MSA and competing algorithms and to show the fastest algorithm to reach the global solution, respectively. The experiments were conducted using MATLAB R2019a on a device with the following specifications: Intel(R) Core(TM) i7-4700MQ CPU @ 2.40 GHz 2.40 GHz, 32 GB RAM, and 64-bit Windows 10 Pro.

4.1. Sensitivity analysis

This section discusses the sensitivity analysis of the proposed algorithm to estimate the optimal value of the MSA's parameters to maximize its performance; these parameters are A , a , p , P , ρ , and P_c . The results pictured in Fig. 7 which are produced from the experiments conducted to estimate these parameters show that the optimal values are 1.0, 0.5, 0.5, 2.0, 6.0, and 0.2 for the parameters A , a , p , P , ρ , and P_c , respectively.

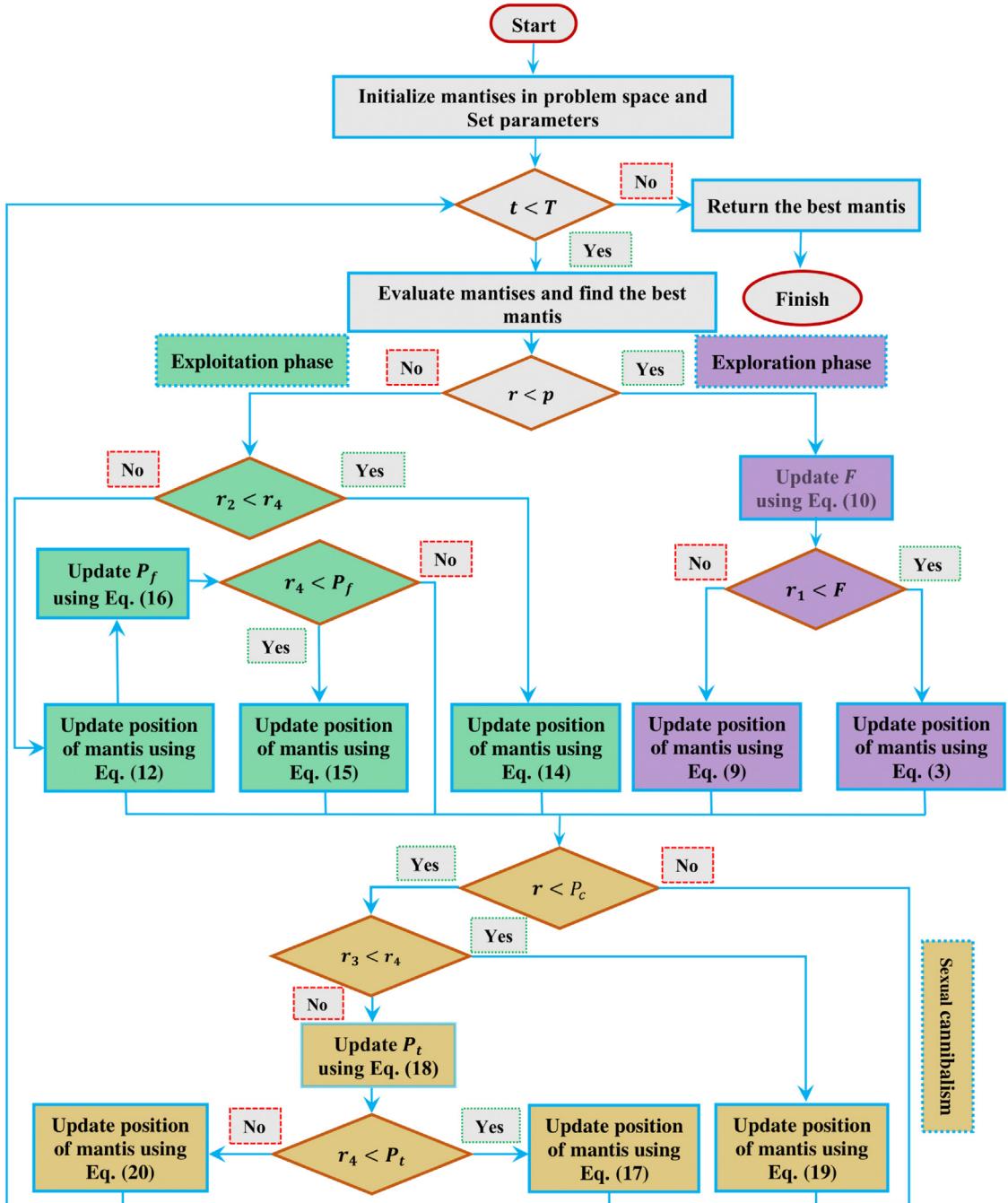


Fig. 6. Flowchart of MSA.

4.2. MSA's evaluation of unimodal test functions

Unimodal test functions have only one global optimal solution, making them ideal for evaluating the exploration ability of optimization algorithms. Thus, in this section, the proposed algorithm's exploitative behavior is evaluated to demonstrate its ability to find the global best solution for these test functions. Table 2 presents the results of the MSA and nine other competing algorithms, such as INFO, GBO, RUN, WOA, GTO, AVOA, DE, GWO, and EO, for

Table 1

Control parameters of optimization algorithms.

Algorithms	Parameters	Value	Algorithms	Parameters	Value
GWO (2014)	Convergence constant α N	Decreases Linearly from 2 to 0 30	DE	Crossover rate Scaling factor	0.5 0.5
WOA (2017)	Convergence constant α Spiral factor b N	Decreases Linearly from 2 to 0 1 30	GTO (2021)	p $Beta$ w N	0.03 3 8 30
EO (2020)	a_1 a_2 V GP N	2 1 1 0.5 30	AVOA (2021)	Alpha (L_1) Beta (L_2) Gamma (w) P_1 P_2 P_3 N	0.8 0.2 2.5 0.6 0.4 0.6 30
RUN (2021)	a b N	20 12 30	INFO (2022)	c d N	2 4 30
GBO (2020)	pr β_{min} β_{max} N	0.5 0.2 1.2 30			

Table 2

Comparison of unimodal test functions.

Function	Metrics	MSA	INFO	GBO	RUN	WOA	GTO	AVOA	DE	GWO	EO
F1	Best	0.000E+00	2.521E-55	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	1.040E-01	7.146E-54	1.633E-112
	Ave	0.000E+00	2.786E-54	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	2.079E-01	3.570E-52	1.478E-108
	Worst	0.000E+00	8.512E-54	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	5.515E-01	1.955E-51	8.756E-108
	SD	0.000E+00	2.420E-54	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	9.478E-02	4.229E-52	2.626E-108
	F-rank	3.500E+00	8.000E+00	3.500E+00	3.500E+00	3.500E+00	3.500E+00	3.500E+00	1.000E+01	9.000E+00	7.000E+00
	Time	3.268E+01	4.352E+01	4.453E+01	8.238E+01	7.149E+00	1.815E+01	1.822E+01	1.515E+01	2.342E+01	1.169E+01
F2	Best	0.000E+00	3.625E-27	1.987E-216	1.705E-201	0.000E+00	0.000E+00	0.000E+00	1.343E-01	1.360E-31	3.173E-64
	Ave	0.000E+00	1.306E-26	1.857E-204	5.206E-184	0.000E+00	0.000E+00	0.000E+00	2.090E-01	4.556E-31	8.343E-63
	Worst	0.000E+00	2.113E-26	4.162E-203	9.347E-183	0.000E+00	0.000E+00	0.000E+00	3.146E-01	9.863E-31	3.249E-62
	SD	0.000E+00	4.272E-27	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	5.310E-02	2.443E-31	8.229E-63
	F-rank	2.500E+00	9.000E+00	5.000E+00	6.000E+00	2.500E+00	2.500E+00	2.500E+00	1.000E+01	8.000E+00	7.000E+00
	Time	3.536E+01	4.623E+01	4.530E+01	8.488E+01	7.285E+00	1.923E+01	1.831E+01	1.526E+01	2.406E+01	1.307E+01
F3	Best	0.000E+00	2.937E-52	0.000E+00	5.103E-307	4.487E+05	0.000E+00	0.000E+00	3.101E+05	1.280E-06	1.153E-15
	Ave	0.000E+00	7.232E-51	0.000E+00	2.603E-264	7.212E+05	0.000E+00	0.000E+00	3.983E+05	5.238E-03	5.932E-10
	Worst	0.000E+00	3.264E-50	0.000E+00	6.508E-263	9.946E+05	0.000E+00	0.000E+00	4.728E+05	5.855E-02	9.388E-09
	SD	0.000E+00	7.485E-51	0.000E+00	0.000E+00	1.369E+05	0.000E+00	0.000E+00	4.439E+04	1.282E-02	1.906E-09
	F-rank	2.500E+00	6.000E+00	2.500E+00	5.000E+00	1.000E+01	2.500E+00	2.500E+00	9.000E+00	8.000E+00	7.000E+00
	Time	9.234E+01	1.074E+02	1.062E+02	2.065E+02	6.316E+01	7.770E+01	8.117E+01	7.337E+01	7.992E+01	7.119E+01
F4	Best	0.000E+00	4.404E-29	3.555E-192	5.874E-175	1.796E+01	0.000E+00	0.000E+00	9.183E+01	5.304E-09	2.203E-21
	Ave	0.000E+00	3.710E-28	1.408E-180	5.839E-156	7.308E+01	0.000E+00	0.000E+00	5.966E+01	2.047E-06	2.121E-17
	Worst	0.000E+00	9.655E-28	3.470E-179	1.354E-154	9.660E+01	0.000E+00	0.000E+00	9.740E+01	4.581E-05	3.372E-16
	SD	0.000E+00	2.274E-28	0.000E+00	2.704E-155	2.544E+01	0.000E+00	0.000E+00	1.246E+00	9.121E-06	6.948E-17
	F-rank	2.000E+00	6.000E+00	4.000E+00	5.000E+00	9.120E+00	2.000E+00	2.000E+00	9.880E+00	8.000E+00	7.000E+00
	Time	3.301E+01	4.510E+01	4.457E+01	8.349E+01	6.898E+00	1.852E+01	1.755E+01	1.483E+01	2.339E+01	1.202E+01
F5	Best	9.102E+01	8.735E+01	8.760E+01	9.407E+01	9.628E+01	3.696E-05	9.587E-05	6.468E+02	9.586E+01	9.320E+01
	Ave	9.278E+01	8.980E+01	9.193E+01	9.669E+01	9.700E+01	1.987E-02	1.471E-03	1.097E+03	9.749E+01	9.419E+01
	Worst	9.449E+01	9.432E+01	9.785E+01	9.827E+01	9.816E+01	2.290E-01	4.777E-03	2.123E+03	9.845E+01	9.713E+01
	SD	9.241E+01	1.690E+01	3.006E+00	1.395E+00	5.392E+01	4.559E-02	1.264E-03	3.077E+02	8.163E-01	8.548E-01
	F-rank	4.640E+00	3.480E+00	4.520E+00	7.520E+00	7.840E+00	1.760E+00	1.240E-00	1.000E+01	8.280E+00	5.720E+00
	Time	3.591E+01	5.051E+01	4.763E+01	9.030E+01	9.341E+01	2.014E+01	2.012E+01	1.876E+01	2.649E+01	1.422E+01
F6	Best	1.073E-26	1.511E-07	1.570E-05	1.857E-07	3.308E-01	3.381E-05	2.707E-07	8.391E-02	7.628E+00	2.338E-04
	Ave	3.081E-12	2.005E-02	5.544E-05	1.102E-06	7.375E-01	6.574E-04	1.240E-06	1.763E-01	9.509E+00	1.901E-01
	Worst	7.645E-11	2.500E-01	1.599E-04	6.587E-06	1.372E+00	4.783E-03	2.178E-06	4.283E-01	1.196E+01	7.496E-01
	SD	1.529E-11	6.917E-02	3.890E-05	1.766E-06	3.012E-01	9.244E-04	6.000E-07	7.505E-02	9.040E-01	2.074E-01
	F-rank	1.000E+00	4.360E+00	4.840E+00	2.320E+00	8.920E+00	5.960E+00	2.920E+00	7.360E+00	1.000E+01	7.320E+00
	Time	3.409E+01	4.677E+01	4.507E+01	8.407E+01	7.294E+00	1.844E+01	1.798E+01	1.621E+01	2.517E+01	1.227E+01

Bold values represent the best outcomes.

six unimodal test functions. The results include the best, average (Ave), worst, F-rank, time, and SD of 30 fitness values. Examining this table reveals that MSA is competitive with GTO and AVOA for the first four unimodal test functions, outperforms all rival algorithms for F6, and performs weaker than AVOA for F5. Regarding the computational cost, MSA is superior to some algorithms such as GBO, INFO, and RUN but inferior to others.

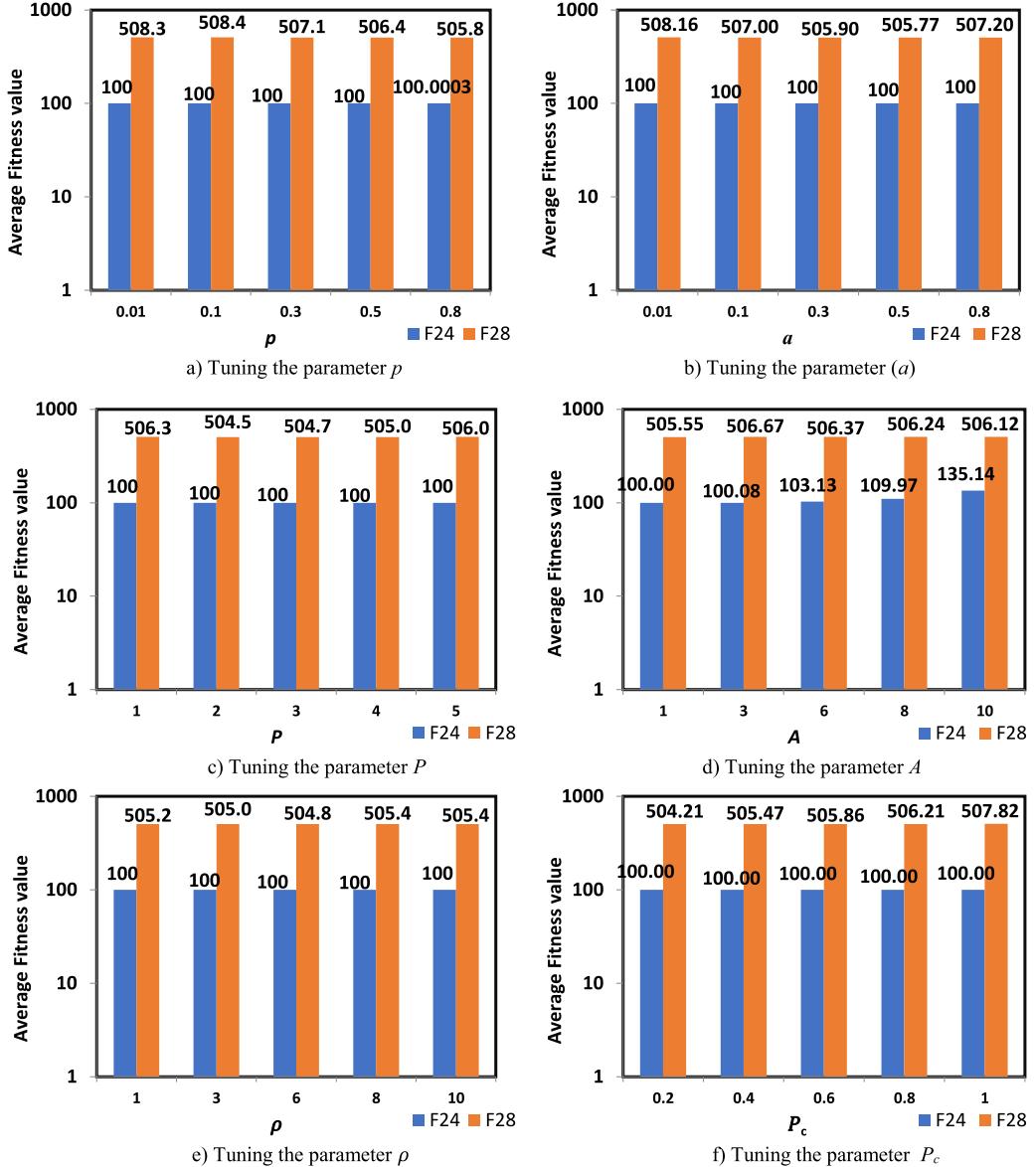


Fig. 7. Tuning control parameters of MSA on F24 and F28 in 10 dimensions (small-scale).

4.3. MSA's evaluation of multimodal test functions

This section evaluates the exploration capability of MSA on multimodal test functions (fixed-dimensional and high-dimensional), as they have multiple local minima. Tables 3 and 4 display the best, average (Ave), worst, F-rank, and SD values of the fitness values obtained from 30 independent runs of the proposed and compared optimizers on both high-dimensional and fixed-dimensional multimodal test functions. The results in those tables indicate that MSA is competitive in terms of the best, Ave, worst, and SD values with some of the compared optimizers for 11 multimodal test functions, performs poorly for three functions (F7, F8, and F13), and surpasses all rival methods for F12, F15, and F20. It is worth noting that MSA could perform better than all rival algorithms in terms of the best metric for F13 and be competitive with them on F8. This comparison confirms that MSA has a strong exploration operator, as it demonstrates various exploration behaviors at different stages of the optimization process.

Table 3

Comparison on high-dimensional multimodal test functions.

Function	Metric	MSA	INFO	GBO	RUN	WOA	GTO	AVOA	DE	GWO	EO
F7	Best	3.875E-05	2.502E-05	2.578E-05	1.565E-05	9.264E-06	1.372E-06	2.327E-06	1.528E-01	6.308E-04	2.392E-04
	Ave	2.583E-04	4.810E-04	2.468E-04	1.589E-04	1.375E-03	5.777E-05	4.085E-05	2.328E-01	1.426E-03	5.549E-04
	Worst	6.391E-04	1.377E-03	1.059E-03	4.017E-04	9.782E-03	1.432E-04	8.532E-05	3.255E-01	2.592E-03	1.166E-03
	SD	1.764E-04	3.392E-04	2.583E-04	8.194E-05	2.109E-03	3.910E-05	2.364E-05	4.726E-02	5.384E-04	2.646E-04
	F-rank	4.920E+00	6.240E+00	4.720E+00	3.840E+00	6.240E+00	1.920E+00	1.680E+00	1.000E+01	8.600E+00	6.840E+00
	Time	5.351E+01	6.859E+01	6.586E+01	1.266E+02	2.634E+01	3.841E+01	3.828E+01	3.557E+01	4.357E+01	3.271E+01
F8	Best	-4.190E+04	-2.867E+04	-3.147E+04	-2.718E+04	-4.190E+04	-4.190E+04	-4.190E+04	-1.517E+04	-2.101E+04	-3.116E+04
	Ave	-3.659E+04	-2.594E+04	-2.861E+04	-2.394E+04	-3.947E+04	-4.190E+04	-4.017E+04	-1.357E+04	-1.680E+04	-2.912E+04
	Worst	-2.105E+04	-2.355E+04	-2.587E+04	-1.900E+04	-2.646E+04	-4.190E+04	-3.451E+04	-1.271E+04	-1.395E+04	-2.498E+04
	SD	7.042E+03	1.367E+03	1.454E+03	2.248E+03	4.303E+03	1.135E-01	2.329E+03	6.228E+02	1.686E+03	1.459E+03
	F-rank	2.640E+00	6.920E+00	5.440E+00	7.880E+00	3.160E+00	1.680E+00	3.040E+00	1.000E+01	8.960E+00	5.280E+00
	Time	3.893E+01	5.266E+01	5.068E+01	9.573E+01	1.151E+01	2.363E+01	2.414E+01	2.074E+01	2.892E+01	1.738E+01
F9	Best	0.000E+00	7.701E+02	0.000E+00	0.000E+00						
	Ave	0.000E+00	8.536E+02	1.364E-14	0.000E+00						
	Worst	0.000E+00	8.942E+02	1.137E-13	0.000E+00						
	SD	0.000E+00	2.977E+01	3.771E-14	0.000E+00						
	F-rank	4.940E+00	1.000E+01	5.480E+00	4.940E+00						
	Time	3.450E+01	5.077E+01	4.675E+01	8.802E+01	8.099E+00	1.967E+01	1.884E+01	2.004E+01	2.497E+01	1.357E+01
F10	Best	8.882E-16	5.809E-02	2.931E-14	4.441E-15						
	Ave	8.882E-16	8.882E-16	8.882E-16	8.882E-16	2.593E-15	8.882E-16	8.882E-16	9.084E-02	3.457E-14	6.857E-15
	Worst	8.882E-16	8.882E-16	8.882E-16	8.882E-16	4.441E-15	8.882E-16	8.882E-16	1.437E-01	3.997E-14	7.994E-15
	SD	0.000E+00	0.000E+00	0.000E+00	0.000E+00	1.812E-15	0.000E+00	0.000E+00	2.109E-02	3.988E-15	1.691E-15
	F-rank	3.760E+00	3.760E+00	3.760E+00	3.760E+00	5.500E+00	3.760E+00	3.760E+00	1.000E+01	9.000E+00	7.940E+00
	Time	3.497E+01	4.829E+01	4.736E+01	8.920E+01	8.622E+00	1.940E+01	1.924E+01	2.049E+01	2.614E+01	1.358E+01
F11	Best	0.000E+00	5.319E-02	0.000E+00	0.000E+00						
	Ave	0.000E+00	9.179E-02	1.272E-03	0.000E+00						
	Worst	0.000E+00	1.323E-01	1.916E-02	0.000E+00						
	SD	0.000E+00	2.280E-02	4.502E-03	0.000E+00						
	F-rank	4.960E+00	1.000E+01	5.320E+00	4.960E+00						
	Time	3.742E+01	5.468E+01	5.069E+01	9.574E+01	1.114E+01	2.200E+01	2.161E+01	2.312E+01	2.899E+01	1.616E+01
F12	Best	4.712E-33	2.033E-09	4.659E-07	9.916E-09	3.870E-03	2.452E-09	9.231E-10	5.487E+00	1.545E-01	4.300E-06
	Ave	1.891E-32	1.573E-04	1.513E-06	2.234E-08	6.987E-03	2.052E-06	5.703E-09	8.260E+01	2.392E-01	8.456E-04
	Worst	3.503E-31	1.966E-03	4.088E-06	7.551E-08	1.323E-02	6.977E-06	1.074E-08	1.701E+03	4.361E-01	3.831E-03
	SD	6.905E-32	5.441E-04	9.241E-07	1.549E-08	2.480E-03	1.945E-06	2.627E-09	3.372E+02	5.854E-02	1.113E-03
	F-rank	1.000E+00	3.840E+00	5.360E+00	3.520E+00	8.000E+00	5.280E+00	2.120E+00	1.000E+01	9.000E+00	6.880E+00
	Time	8.987E+01	1.042E+02	9.947E+01	1.943E+02	5.870E+01	6.889E+01	6.878E+01	7.031E+01	7.601E+01	6.292E+01
F13	Best	1.350E-32	4.440E-01	1.127E-02	1.025E-06	4.985E-01	3.316E-08	6.551E-08	3.999E+01	5.555E+00	4.148E-01
	Ave	1.946E+00	1.982E+00	2.860E+00	3.833E-02	1.130E+00	7.858E-05	5.620E-07	1.063E+02	6.050E+00	2.774E+00
	Worst	8.016E+00	9.886E+00	6.788E+00	1.881E-01	2.309E+00	3.639E-04	3.592E-06	3.789E+02	6.731E+00	5.405E+00
	SD	3.180E+00	2.190E+00	2.181E+00	4.816E-02	5.306E-02	1.101E-01	7.015E-07	7.289E+01	3.392E+01	1.313E+00
	F-rank	4.040E+00	6.040E+00	6.280E+00	3.720E+00	5.400E+00	2.360E+00	1.520E+00	1.000E+01	8.640E+00	7.000E+00
	Time	8.826E+01	1.028E+02	9.825E+01	1.932E+02	5.453E+01	8.943E+01	8.759E+01	7.014E+01	7.424E+01	6.248E+01
F14	Best	9.980E-01	9.980E-01	9.980E-01	9.980E-01	9.980E-01	9.980E-01	9.980E-01	9.980E-01	9.980E-01	9.980E-01
	Ave	9.980E-01	1.859E+00	9.980E-01	4.023E+00	1.474E+00	9.980E-01	9.980E-01	9.980E-01	5.142E+00	9.980E-01
	Worst	9.980E-01	1.076E+01	9.980E-01	1.076E+01	2.982E+00	9.980E-01	9.980E-01	9.980E-01	1.267E+01	9.980E-01
	SD	0.000E+00	2.709E+00	0.000E+00	3.525E+00	8.648E-01	4.532E-17	2.398E-16	0.000E+00	4.718E+00	6.410E-17
	F-rank	3.620E+00	4.340E+00	3.620E+00	9.000E+00	8.520E+00	3.760E+00	5.440E+00	3.620E+00	9.000E+00	3.880E+00
	Time	5.541E+01	7.641E+01	6.857E+01	1.395E+02	4.311E+01	4.687E+01	4.634E+01	5.160E+01	4.302E+01	4.483E+01
F15	Best	3.075E-04	3.075E-04	3.075E-04	3.075E-04	3.077E+01	3.075E-04	3.075E-04	3.075E-04	3.075E-04	3.075E-04
	Ave	3.075E-04	3.807E-04	4.906E-04	6.788E-04	6.930E-04	3.075E-04	3.075E-04	3.075E-04	1.268E-03	1.949E-03
	Worst	3.075E-04	1.223E-03	1.223E-03	1.223E-03	1.489E-03	3.075E-04	3.075E-04	3.075E-04	2.036E-02	2.036E-02
	SD	1.032E-19	2.535E-04	3.738E-04	4.543E-04	4.042E-04	2.623E-19	2.041E-08	3.993E-03	8.102E-03	5.545E-03
	F-rank	2.780E+00	2.320E+00	4.180E+00	7.120E+00	8.800E+00	3.820E+00	6.520E+00	4.060E+00	8.000E+00	7.400E+00
	Time	1.567E+01	3.077E+01	2.652E+01	5.453E+01	4.531E+00	8.943E+00	8.759E+00	1.112E+01	5.044E+00	6.547E+00
F16	Best	-1.032E+00									
	Ave	-1.032E+00									
	Worst	-1.032E+00									
	SD	6.799E-16	6.799E-16	6.799E-16	6.799E-16	1.676E-13	7.443E-12	6.490E-16	4.965E-16	6.799E-16	5.346E-16
	F-rank	3.420E+00	3.420E+00	3.420E+00	3.420E+00	8.220E+00	4.000E+00	6.400E+00	3.420E+00	1.000E+01	3.980E+00
	Time	1.477E+01	2.912E+01	2.593E+01	5.294E+01	3.995E+00	8.063E+00	8.201E+00	1.034E+01	4.152E+00	6.117E+00
F17	Best	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01
	Ave	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01
	Worst	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01
	SD	0.000E+00	0.000E+00	0.000E+00	0.000E+00	1.889E-11	1.453E-07	0.000E+00	0.000E+00	0.000E+00	0.000E+00
	F-rank	4.060E+00	4.060E+00	4.060E+00	4.060E+00	7.620E+00	9.520E+00	4.060E+00	4.060E+00	4.060E+00	4.060E+00
	Time	1.398E+01	2.799E+01	2.							

Table 4 (continued).

Function	Metric	MSA	INFO	GBO	RUN	WOA	GTO	AVOA	DE	GWO	EO
F19	Best	-3.863E+00	-3.863E+00	-3.863E+00	-3.863E+00	-3.863E+00	-3.863E+00	-3.863E+00	-3.863E+00	-3.863E+00	-3.863E+00
	Ave	-3.863E+00	-3.863E+00	-3.863E+00	-3.863E+00	-3.860E+00	-3.863E+00	-3.863E+00	-3.862E+00	-3.863E+00	-3.863E+00
	Worst	-3.863E+00	-3.863E+00	-3.863E+00	-3.863E+00	-3.854E+00	-3.863E+00	-3.863E+00	-3.855E+00	-3.863E+00	-3.863E+00
	SD	2.266E-15	2.266E-15	2.266E-15	2.266E-15	7.203E-08	3.197E-03	2.211E-15	1.760E-15	2.266E-15	2.678E-03
	F-rank	3.440E+00	3.440E+00	3.440E+00	8.040E+00	9.800E+00	3.720E+00	6.700E+00	3.440E+00	9.160E+00	3.820E+00
	Time	1.553E+01	2.960E+01	2.653E+01	5.504E+01	4.652E+00	8.462E+00	9.002E+00	1.107E+01	4.802E+00	6.590E+00
F20	Best	-3.322E+00	-3.322E+00	-3.322E+00	-3.322E+00	-3.322E+00	-3.322E+00	-3.322E+00	-3.322E+00	-3.322E+00	-3.322E+00
	Ave	-3.322E+00	-3.289E+00	-3.274E+00	-3.274E+00	-3.256E+00	-3.251E+00	-3.270E+00	-3.271E+00	-3.274E+00	-3.274E+00
	Worst	-3.322E+00	-3.203E+00	-3.203E+00	-3.203E+00	-3.087E+00	-3.203E+00	-3.203E+00	-3.197E+00	-3.203E+00	-3.203E+00
	SD	4.441E-16	5.448E-02	5.945E-02	5.945E-02	8.008E-02	6.062E-02	5.945E-02	6.023E-02	6.073E-02	5.945E-02
	F-rank	2.440E+00	3.920E+00	4.600E+00	7.120E+00	8.080E+00	5.280E+00	6.940E+00	4.640E+00	7.280E+00	4.700E+00
	Time	1.671E+01	3.153E+01	2.744E+01	5.609E+01	4.627E+00	9.460E+00	9.416E+00	1.095E+01	5.442E+00	7.076E+00
F21	Best	-1.015E+01	-1.015E+01	-1.015E+01	-1.015E+01	-1.015E+01	-1.015E+01	-1.015E+01	-1.015E+01	-1.015E+01	-1.015E+01
	Ave	-1.015E+01	-9.852E+00	-7.706E+00	-9.949E+00	-9.026E+00	-1.015E+01	-9.026E+00	-9.353E+00	-9.951E+00	-8.930E+00
	Worst	-1.015E+01	-2.630E+00	-5.055E+00	-5.055E+00	-2.630E+00	-1.015E+01	-1.015E+01	-2.638E+00	-5.101E+00	-5.055E+00
	SD	5.439E-15	1.505E+00	2.599E+00	1.020E+00	2.317E+00	4.514E-15	4.728E-15	2.247E+00	1.010E+00	2.222E+00
	F-rank	2.400E+00	3.000E+00	6.120E+00	7.160E+00	9.200E+00	3.940E+00	5.740E+00	3.520E+00	8.160E+00	5.760E+00
	Time	1.829E+01	3.351E+01	2.876E+01	5.892E+01	6.098E+00	1.069E+01	1.056E+01	1.342E+01	6.639E+00	8.388E+00
F22	Best	-1.040E+01	-1.040E+01	-1.040E+01	-1.040E+01	-1.040E+01	-1.040E+01	-1.040E+01	-1.040E+01	-1.040E+01	-1.040E+01
	Ave	-1.040E+01	-9.563E+00	-8.489E+00	-1.040E+01	-1.019E+01	-1.040E+01	-1.040E+01	-9.735E+00	-1.040E+01	-9.978E+00
	Worst	-1.040E+01	-2.766E+00	-5.088E+00	-1.040E+01	-5.088E+00	-1.040E+01	-1.040E+01	-2.766E+00	-1.040E+01	-5.088E+00
	SD	2.689E-15	2.326E+00	2.604E+00	8.233E-10	1.063E+00	3.516E-15	3.478E-15	2.015E+00	2.545E-05	1.472E+00
	F-rank	2.820E+00	4.420E+00	5.500E+00	7.200E+00	9.360E+00	3.880E+00	5.560E+00	3.780E+00	8.240E+00	4.240E+00
	Time	1.884E+01	3.458E+01	3.012E+01	6.147E+01	6.947E+00	1.172E+01	1.167E+01	1.410E+01	7.665E+00	9.380E+00
F23	Best	-1.054E+01	-1.054E+01	-1.054E+01	-1.054E+01	-1.054E+01	-1.054E+01	-1.054E+01	-1.054E+01	-1.054E+01	-1.054E+01
	Ave	-1.054E+01	-8.985E+00	-9.347E+00	-1.032E+01	-9.883E+00	-1.054E+01	-1.054E+01	-1.021E+01	-1.054E+01	-9.836E+00
	Worst	-1.054E+01	-2.422E+00	-2.422E+00	-5.128E+00	-5.128E+00	-1.054E+01	-1.054E+01	-2.427E+00	-1.054E+01	-3.835E+00
	SD	1.813E-15	3.176E+00	2.479E+00	1.082E+00	1.792E+00	3.223E-15	1.904E-15	1.622E+00	3.354E-05	1.949E+00
	F-rank	3.020E+00	4.420E+00	4.660E+00	7.520E+00	9.360E+00	3.740E+00	6.300E+00	3.220E+00	8.440E+00	4.320E+00
	Time	1.998E+01	3.616E+01	3.137E+01	6.394E+01	8.284E+00	1.264E+01	1.291E+01	1.578E+01	8.877E+00	1.040E+01

Bold values represent the best outcomes.

4.4. Comparison on CEC2017

This section evaluates the performance of MSA and its rival algorithms on the challenging and recent benchmark, CEC-2017 [121]. This benchmark comprises four categories of mathematical test functions: unimodal functions (F24–F25), multimodal functions (F26–F32), composition functions (F33–F42), and hybrid functions (F43–F52). As previously mentioned, unimodal functions are ideal for assessing the exploitative abilities of optimization algorithms since they only have a global best solution, while multimodal test functions, with multiple local optimal solutions, are well-suited for evaluating the exploration capabilities of newly proposed optimization algorithms. The composition and hybrid functions are designed to test the algorithms' ability to avoid local optima. In the experiments discussed in this section, the dimension is set to 10 for all functions in this benchmark. The details of the CEC-2017 benchmark can be found in [121].

The results of the MSA and its rival optimizers, obtained from 30 independent runs on the unimodal and multimodal test functions, are displayed in Tables 5, 6, and 7 in terms of the best, average, worst, F-rank, and SD values of the fitness values. The results demonstrate that MSA is the best optimizer for the unimodal, multimodal, composition, and hybrid test functions, outperforming 23 out of the 29 test functions and being competitive with the others. The average of F-rank and SD values from each column in the last mentioned tables are presented in Figs. 8 and 9, respectively. Inspecting Fig. 8 shows that the means of MSA are better than those of the rival optimizers. MSA is more stable than all the other optimizers, as reported in Fig. 9.

To determine the difference between the outcomes of MSA and its rival algorithms, the Wilcoxon rank-sum test was used to calculate the *p*-value between the outcomes of MSA and each rival algorithm on each CEC-2017 test function. The results of the Wilcoxon rank-sum test, shown in Table 8, confirm that there is a significant difference between the outcomes of MSA and all rival algorithms on most test functions. These experiments validate that MSA is a strong alternative optimizer, outperforming three well-established optimizers (GWO, WOA, and DE) and seven recently-published optimizers (INFO, GBO, RUN, GTO, AVOA, and EO).

4.5. MSA's qualitative analysis

In this section, the performance of MSA during the optimization process is evaluated using five well-known indicators: diversity, convergence curve, average fitness value, trajectory in the first dimension, and search history. The diversity metric represents the average distance between individuals during the optimization process, the convergence curve illustrates the best fitness values achieved at each iteration, the average fitness value displays

Table 5

Comparison of unimodal and multimodal test functions of CEC-2017 benchmark.

Function	Metric	MSA	INFO	GBO	RUN	WOA	GTO	AVOA	DE	GWO	EO
F24	Best	1.000E+02	1.000E+02	1.057E+02	1.456E+02	9.225E+04	1.068E+02	1.131E+02	1.000E+02	1.942E+03	1.000E+02
	Ave	1.000E+02	1.000E+02	1.667E+03	3.214E+03	2.809E+06	1.250E+03	2.457E+03	1.000E+02	5.320E+07	1.773E+03
	Worst	1.000E+02	1.000E+02	6.884E+03	9.876E+03	2.011E+07	3.648E+03	9.267E+03	1.000E+02	4.821E+08	1.217E+04
	SD	1.485E-13	3.642E-09	1.526E+03	1.968E+03	5.187E+06	1.281E+03	2.399E+03	4.511E-12	1.330E+08	2.628E+03
	F-rank	1.120E+00	2.880E+00	5.920E+00	7.160E+00	9.680E+00	5.320E+00	6.280E+00	2.000E+00	9.040E+00	5.600E+00
	Time	1.887E+01	3.872E+01	3.316E+01	6.553E+01	5.779E+00	1.059E+01	1.051E+01	1.285E+01	7.256E+00	8.024E+00
F25	Best	3.000E+02	3.000E+02	3.000E+02	3.000E+02	4.851E+02	3.000E+02	3.000E+02	3.000E+02	3.067E+02	3.000E+02
	Ave	3.000E+02	3.000E+02	3.000E+02	3.000E+02	1.969E+03	3.000E+02	3.000E+02	3.000E+02	2.153E+03	3.000E+02
	Worst	3.000E+02	3.000E+02	3.000E+02	3.000E+02	5.212E+03	3.000E+02	3.000E+02	3.000E+02	9.467E+03	3.000E+02
	SD	2.321E-14	3.647E-13	2.507E-13	4.969E-04	1.545E+03	1.791E-10	3.149E-09	1.730E-04	2.612E+03	1.911E-10
	F-rank	1.160E+00	2.880E+00	2.560E+00	7.960E+00	9.600E+00	3.840E+00	5.960E+00	7.040E+00	9.400E+00	4.600E+00
	Time	1.933E+01	3.881E+01	3.347E+01	6.609E+01	6.252E+00	1.113E+01	1.085E+01	1.333E+01	7.733E+00	8.608E+00
F26	Best	4.000E+02	4.000E+02	4.000E+02	4.000E+02	4.045E+02	4.000E+02	4.001E+02	4.030E+02	4.006E+02	4.000E+02
	Ave	4.000E+02	4.000E+02	4.000E+02	4.000E+02	4.028E+02	4.436E+02	4.018E+02	4.093E+02	4.044E+02	4.039E+02
	Worst	4.000E+02	4.001E+02	4.032E+02	4.054E+02	5.450E+02	4.036E+02	4.668E+02	4.057E+02	5.371E+02	4.048E+02
	SD	8.462E-05	3.201E-02	6.276E-01	2.321E+00	4.181E+01	1.190E+00	1.747E+01	8.037E-01	3.178E+01	9.354E-01
	F-rank	1.160E+00	2.480E+00	2.720E+00	5.880E+00	9.360E+00	4.640E+00	6.880E+00	6.640E+00	9.200E+00	6.040E+00
	Time	1.903E+01	3.931E+01	3.363E+01	6.639E+01	6.274E+00	1.103E+01	1.095E+01	1.376E+01	7.767E+00	8.571E+00
F27	Best	5.020E+02	5.090E+02	5.080E+02	5.139E+02	5.230E+02	5.080E+02	5.139E+02	5.076E+02	5.039E+02	5.006E+02
	Ave	5.047E+02	5.200E+02	5.246E+02	5.311E+02	5.453E+02	5.244E+02	5.347E+02	5.168E+02	5.125E+02	5.125E+02
	Worst	5.080E+02	5.378E+02	5.557E+02	5.716E+02	5.787E+02	5.478E+02	5.617E+02	5.214E+02	5.325E+02	5.308E+02
	SD	1.647E+00	7.440E+00	1.030E+01	1.198E+01	1.607E+01	9.983E+00	1.256E+01	3.298E+00	7.581E+00	5.536E+00
	F-rank	1.080E+00	5.040E+00	6.260E+00	7.680E+00	9.200E+00	6.140E+00	8.000E+00	4.360E+00	4.280E+00	3.080E+00
	Time	1.964E+01	3.956E+01	3.450E+01	6.806E+01	6.763E+00	1.154E+01	1.152E+01	1.412E+01	8.351E+00	9.197E+00
F28	Best	6.000E+02	6.000E+02	6.000E+02	6.065E+02	6.173E+02	6.003E+02	6.006E+02	6.000E+02	6.000E+02	6.000E+02
	Ave	6.000E+02	6.011E+02	6.009E+02	6.177E+02	6.360E+02	6.058E+02	6.101E+02	6.000E+02	6.013E+02	6.000E+02
	Worst	6.000E+02	6.088E+02	6.160E+02	6.331E+02	6.692E+02	6.251E+02	6.285E+02	6.000E+02	6.091E+02	6.000E+02
	SD	5.684E-14	2.211E+00	3.193E+00	7.679E+00	1.276E+01	5.069E+00	8.407E+00	2.321E-14	2.373E+00	1.445E-03
	F-rank	1.600E+00	4.960E+00	4.920E+00	8.760E+00	9.760E+00	6.880E+00	8.000E+00	1.400E+00	5.640E+00	3.080E+00
	Time	2.201E+01	4.198E+01	3.671E+01	7.240E+01	8.796E+00	1.365E+01	1.352E+01	1.617E+01	1.043E+01	1.116E+01
F29	Best	7.127E+02	7.168E+02	7.204E+02	7.224E+02	7.519E+02	7.214E+02	7.471E+02	7.246E+02	7.131E+02	7.149E+02
	Ave	7.165E+02	7.383E+02	7.417E+02	7.578E+02	7.894E+02	7.456E+02	7.736E+02	7.295E+02	7.206E+02	7.206E+02
	Worst	7.207E+02	7.831E+02	7.784E+02	7.871E+02	8.465E+02	7.834E+02	8.028E+02	7.350E+02	7.497E+02	7.346E+02
	SD	2.203E+00	1.724E+01	1.334E+01	1.432E+01	2.269E+01	1.735E+01	1.611E+01	2.584E+00	8.528E+00	4.945E+00
	F-rank	1.320E+00	5.200E+00	6.080E+00	7.960E+00	9.320E+00	6.000E+00	8.720E+00	4.120E+00	4.040E+00	2.240E+00
	Time	1.995E+01	4.038E+01	3.545E+01	6.986E+01	7.220E+00	1.201E+01	1.202E+01	1.456E+01	8.559E+00	9.443E+00
F30	Best	8.010E+02	8.060E+02	8.070E+02	8.199E+02	8.151E+02	8.080E+02	8.060E+02	8.099E+02	8.080E+02	8.050E+02
	Ave	8.050E+02	8.198E+02	8.225E+02	8.265E+02	8.420E+02	8.231E+02	8.254E+02	8.177E+02	8.157E+02	8.130E+02
	Worst	8.109E+02	8.448E+02	8.497E+02	8.398E+02	8.886E+02	8.378E+02	8.458E+02	8.247E+02	8.358E+02	8.229E+02
	SD	2.457E+00	8.974E+00	1.038E+01	4.710E+00	1.672E+01	8.266E+00	1.041E+01	3.756E+00	6.481E+00	4.999E+00
	F-rank	1.120E+00	5.440E+00	5.940E+00	7.480E+00	9.080E+00	6.600E+00	6.960E+00	4.720E+00	4.080E+00	3.580E+00
	Time	1.984E+01	3.993E+01	3.451E+01	6.831E+01	6.915E+00	1.156E+01	1.163E+01	1.424E+01	8.466E+00	9.364E+00
F31	Best	9.000E+02	9.000E+02	9.000E+02	9.537E+02	9.530E+02	9.001E+02	9.035E+02	9.000E+02	9.001E+02	9.000E+02
	Ave	9.000E+02	9.170E+02	9.086E+02	1.043E+03	1.406E+03	9.370E+02	1.036E+03	9.000E+02	9.144E+02	9.005E+02
	Worst	9.000E+02	9.840E+02	1.002E+03	1.227E+03	2.146E+03	1.102E+03	1.707E+03	9.000E+02	9.547E+02	9.089E+02
	SD	0.000E+00	2.362E+01	2.136E+01	6.201E+01	2.717E+02	4.627E+01	1.696E+02	0.000E+00	1.744E+01	1.837E+00
	F-rank	1.700E+00	5.300E+00	4.620E+00	8.600E+00	9.720E+00	6.360E+00	8.240E+00	1.700E+00	5.480E+00	3.280E+00
	Time	2.022E+01	4.062E+01	3.511E+01	6.953E+01	7.238E+00	1.190E+01	1.193E+01	1.467E+01	8.634E+00	9.594E+00
F32	Best	1.000E+03	1.246E+03	1.239E+03	1.257E+03	1.553E+03	1.245E+03	1.252E+03	1.142E+03	1.135E+03	1.032E+03
	Ave	1.228E+03	1.752E+03	1.849E+03	1.773E+03	2.020E+03	1.767E+03	1.911E+03	1.611E+03	1.638E+03	1.511E+03
	Worst	1.592E+03	2.256E+03	2.485E+03	2.196E+03	2.476E+03	2.280E+03	2.466E+03	2.098E+03	2.507E+03	1.916E+03
	SD	1.593E+02	2.953E+02	3.277E+02	2.364E+02	2.827E+02	2.614E+02	3.136E+02	2.511E+02	3.458E+02	2.407E+02
	F-rank	1.560E+00	5.920E+00	6.600E+00	6.200E+00	7.960E+00	6.160E+00	7.440E+00	4.640E+00	4.720E+00	3.800E+00
	Time	2.062E+01	4.059E+01	3.551E+01	6.992E+01	7.353E+00	1.218E+01	1.206E+01	1.473E+01	8.924E+00	9.810E+00
F33	Best	1.100E+03	1.102E+03	1.100E+03	1.110E+03	1.117E+03	1.105E+03	1.108E+03	1.100E+03	1.106E+03	1.100E+03
	Ave	1.101E+03	1.129E+03	1.114E+03	1.123E+03	1.285E+03	1.128E+03	1.157E+03	1.101E+03	1.157E+03	1.107E+03
	Worst	1.105E+03	1.199E+03	1.168E+03	1.134E+03	2.650E+03	1.163E+03	1.527E+03	1.104E+03	1.362E+03	1.123E+03
	SD	1.466E+00	3.211E+01	1.443E+01	6.203E+00	3.068E+02	1.467E+01	8.222E+01	1.227E+00	6.209E+01	4.977E+00
	F-rank	1.680E+00	5.840E+00	4.480E+00	6.440E+00	9.400E+00	6.880E+00	7.640E+00	4.720E+00	7.440E+00	3.480E+00
	Time	2.001E+01	4.067E+01	3.519E+01	6.971E+01	6.758E+00	1.162E+01	1.156E+01	1.412E+01	8.306E+00	9.143E+00
F34	Best	1.201E+03	1.354E+03	1.563E+03	2.791E+03	4.712E+03	1.653E+03	2.023E+04	1.860E+03	1.439E+04	1.775E+03
	Ave	1.487E+03	4.227E+03	1.122E+04	1.566E+04	3.975E+06	8.512E+03	3.819E+05	2.851E+03	8.826E+05	1.258E+04
	Worst	3.719E+03	3.513E+04	5.198E+04	5.538E+04	2.070E+07	4.086E+04	4.183E+06	1.181E+04	2.888E+06	3.984E+04
	SD	4.792E+02	6.779E+03	1.215E+04	1.356E+04	5.552E+06	8.640E+03	3.724E+05	1.934E+03	8.472E+05	9.791E+03
	F-rank	1.240E+00	2.800E+00	5.160E+00	6.000E+00	8.960E+00	6.480E+00	8.640E+00	3.080E+00	8.920E+00	5.520E+00
	Time	2.000E+01	4.102E+01	3.505E+01	6.966E+01	6.890E+00	1.177E+01	1.190E+01	1.449E+01	8.376E+00	9.297E+00
F35	Best	1.301E+03	1.305E+03	1.331E+03	2.404E+03	2.432E+03	1.315E+03	1.666E+03	1.306E+03	2.405E+03	1.406E+03
	Ave	1.306E+03	1.410E+03	1.722E+03	9.180E+03	1.778E+04	1.472E+03	8.187E+03	1.308E+03	9.767E+03	6.691E+03
	Worst	1.311E+03	2.021E+03	2.498E+03	2.098E+04	5.749E+04	1.792E+03	2.572E+04	1.311E+03	3.460E+04	2.002E+04
	SD	3.129E+00	1.632E+02	3.382E+02	5.057E+03	1					

Table 6 (continued).

Function	Metric	MSA	INFO	GBO	RUN	WOA	GTO	AVOA	DE	GWO	EO
F36	Best	1.400E+03	1.404E+03	1.428E+03	1.441E+03	1.460E+03	1.414E+03	1.441E+03	1.400E+03	1.443E+03	1.427E+03
	Ave	1.402E+03	1.437E+03	1.482E+03	1.747E+03	2.083E+03	1.446E+03	1.656E+03	1.404E+03	3.042E+03	1.459E+03
	Worst	1.404E+03	1.482E+03	1.569E+03	2.248E+03	5.099E+03	1.475E+03	4.006E+03	1.421E+03	5.643E+03	1.524E+03
	SD	9.438E-01	1.466E+01	3.616E+01	3.160E+02	1.098E+03	1.537E+01	5.292E+02	4.696E+00	1.828E+03	2.633E+01
	F-rank	1.320E+00	3.600E+00	6.400E+00	8.040E+00	8.400E+00	4.560E+00	7.400E+00	1.680E+00	8.280E+00	5.320E+00
	Time	2.066E+01	4.132E+01	3.582E+01	7.061E+01	7.313E+00	1.213E+01	1.203E+01	1.478E+01	8.870E+00	9.771E+00
F37	Best	1.500E+03	1.501E+03	1.505E+03	1.520E+03	1.793E+03	1.502E+03	1.530E+03	1.500E+03	1.586E+03	1.519E+03
	Ave	1.501E+03	1.543E+03	1.607E+03	1.754E+03	8.869E+03	1.536E+03	3.404E+03	1.500E+03	4.606E+03	1.606E+03
	Worst	1.502E+03	1.628E+03	1.858E+03	3.362E+03	2.470E+04	1.601E+03	8.146E+03	1.501E+03	2.391E+04	1.754E+03
	SD	5.454E-01	3.727E+01	9.247E+01	4.412E+02	5.736E+03	2.272E+01	1.969E+03	2.781E-01	4.461E+03	5.723E+01
	F-rank	1.800E+00	4.240E+00	5.680E+00	5.760E+00	9.720E+00	4.200E+00	7.960E+00	1.200E+00	8.520E+00	5.920E+00
	Time	1.973E+01	4.064E+01	3.525E+01	6.926E+01	6.719E+00	1.148E+01	1.150E+01	1.416E+01	8.230E+00	9.112E+00
F38	Best	1.600E+03	1.600E+03	1.600E+03	1.605E+03	1.608E+03	1.602E+03	1.605E+03	1.600E+03	1.614E+03	1.601E+03
	Ave	1.600E+03	1.691E+03	1.757E+03	1.775E+03	1.908E+03	1.714E+03	1.799E+03	1.600E+03	1.760E+03	1.652E+03
	Worst	1.601E+03	1.886E+03	2.008E+03	1.995E+03	2.188E+03	1.959E+03	1.993E+03	1.638E+03	1.975E+03	1.871E+03
	SD	2.279E-01	7.790E+01	1.131E+02	1.261E+02	1.606E+02	9.744E+01	1.003E+02	8.181E+00	1.067E+02	6.642E+01
	F-rank	1.520E+00	5.040E+00	6.480E+00	6.720E+00	8.840E+00	5.320E+00	7.520E+00	1.920E+00	7.200E+00	4.440E+00
	Time	2.028E+01	4.089E+01	3.539E+01	6.997E+01	7.057E+00	1.184E+01	1.181E+01	1.449E+01	8.526E+00	9.941E+00
F39	Best	1.700E+03	1.712E+03	1.722E+03	1.737E+03	1.744E+03	1.715E+03	1.715E+03	1.700E+03	1.722E+03	1.700E+03
	Ave	1.702E+03	1.745E+03	1.754E+03	1.757E+03	1.806E+03	1.735E+03	1.758E+03	1.704E+03	1.758E+03	1.742E+03
	Worst	1.706E+03	1.844E+03	1.859E+03	1.804E+03	1.909E+03	1.755E+03	1.824E+03	1.729E+03	1.899E+03	1.823E+03
	SD	1.328E+00	3.042E+01	3.478E+01	1.379E+01	5.009E+01	9.220E+00	2.485E+01	7.622E+00	3.852E+01	2.646E+01
	F-rank	1.880E+00	5.400E+00	6.200E+00	7.480E+00	9.000E+00	4.800E+00	6.960E+00	1.200E+00	6.600E+00	5.480E+00
	Time	2.221E+01	4.282E+01	3.751E+01	7.404E+01	8.804E+00	1.364E+01	1.369E+01	1.645E+01	1.046E+01	1.138E+01
F40	Best	1.800E+03	1.821E+03	1.896E+03	2.320E+03	3.317E+03	1.812E+03	2.705E+03	1.800E+03	2.459E+03	2.026E+03
	Ave	1.801E+03	1.842E+03	3.577E+03	1.521E+04	1.569E+04	2.276E+03	1.157E+04	1.802E+03	2.907E+04	9.505E+03
	Worst	1.803E+03	1.967E+03	2.013E+04	4.813E+04	3.477E+04	8.428E+03	3.308E+04	1.820E+03	5.525E+04	3.476E+04
	SD	6.658E-01	2.847E+01	3.903E+03	1.223E+04	9.091E+03	1.365E+03	8.567E+03	5.521E+00	1.597E+04	1.097E+04
	F-rank	1.880E+00	3.200E+00	5.280E+00	7.920E+00	8.080E+00	4.160E+00	7.600E+00	1.120E+00	9.160E+00	6.600E+00
	Time	2.019E+01	4.088E+01	3.540E+01	6.999E+01	7.129E+00	1.178E+01	1.185E+01	1.451E+01	8.533E+00	9.418E+00
F41	Best	1.900E+03	1.902E+03	1.915E+03	1.907E+03	2.745E+03	1.905E+03	1.919E+03	1.900E+03	1.923E+03	1.910E+03
	Ave	1.900E+03	1.916E+03	1.973E+03	8.638E+03	3.247E+04	1.917E+03	5.752E+03	1.900E+03	5.916E+03	1.962E+03
	Worst	1.910E+03	1.964E+03	2.090E+03	2.166E+04	1.329E+05	1.944E+03	1.617E+04	1.900E+03	1.623E+04	2.099E+03
	SD	2.695E-01	1.516E+01	5.039E+01	6.655E+03	3.918E+04	1.151E+01	4.078E+03	1.088E-02	5.736E+03	4.588E+01
	F-rank	1.880E+00	3.880E+00	6.440E+00	7.160E+00	9.360E+00	4.040E+00	7.680E+00	1.120E+00	7.400E+00	6.040E+00
	Time	3.114E+01	5.113E+01	4.578E+01	9.067E+01	1.722E+01	2.240E+01	2.246E+01	2.497E+01	1.878E+01	2.005E+01
F42	Best	2.000E+03	2.003E+03	2.008E+03	2.030E+03	2.052E+03	2.013E+03	2.022E+03	2.000E+03	2.024E+03	2.001E+03
	Ave	2.000E+03	2.030E+03	2.080E+03	2.097E+03	2.156E+03	2.038E+03	2.067E+03	2.001E+03	2.081E+03	2.045E+03
	Worst	2.001E+03	2.136E+03	2.291E+03	2.196E+03	2.330E+03	2.091E+03	2.203E+03	2.017E+03	2.218E+03	2.162E+03
	SD	3.284E-01	2.572E+01	7.472E+01	5.833E+01	6.565E+01	1.766E+01	4.697E+01	3.328E+00	5.768E+01	5.184E+01
	F-rank	1.440E+00	4.240E+00	6.820E+00	7.640E+00	9.040E+00	5.320E+00	6.600E+00	1.580E+00	7.280E+00	5.040E+00
	Time	2.254E+01	4.304E+01	3.733E+01	7.387E+01	8.929E+00	1.375E+01	1.376E+01	1.626E+01	1.054E+01	1.143E+01

Bold values represent the best outcomes.

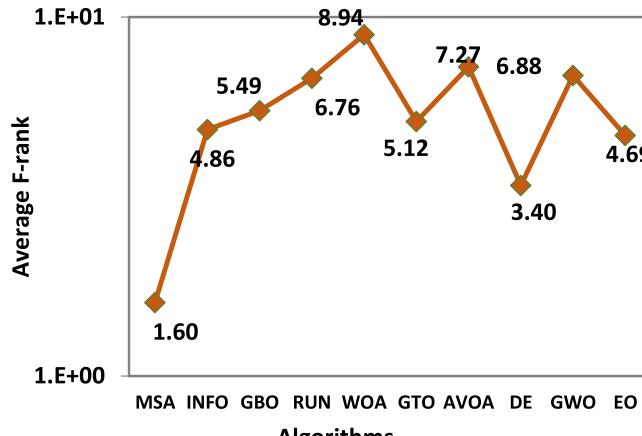


Fig. 8. Average F-rank on all selected CEC-2017 functions.

the average of the fitness values of all individuals at each iteration to demonstrate the impact of the optimization process on each individual's fitness, the trajectory curve shows how the first dimension of a solution evolves over the optimization process, and the search history provides insight into the positions achieved by MSA during the optimization process.

Regarding the diversity metric shown within the second column in Fig. 10, MSA has a trend that gradually decreases during the optimization process to shift the performance of MSA from exploration to exploitation. For different types of test functions, MSA maintains population diversity in the initial stage of the optimization process

Table 7

Comparison of composition test functions of CEC-2017 benchmark.

Function	Metric	MSA	INFO	GBO	RUN	WOA	GTO	AVOA	DE	GWO	EO
F43	Best	2.200E+03	2.200E+03	2.200E+03	2.200E+03	2.211E+03	2.200E+03	2.200E+03	2.203E+03	2.205E+03	2.200E+03
	Ave	2.204E+03	2.315E+03	2.248E+03	2.253E+03	2.318E+03	2.200E+03	2.212E+03	2.313E+03	2.309E+03	2.298E+03
	Worst	2.307E+03	2.350E+03	2.350E+03	2.331E+03	2.377E+03	2.203E+03	2.331E+03	2.332E+03	2.335E+03	2.322E+03
	SD	2.132E+01	3.677E+01	6.319E+01	6.052E+01	5.361E+01	9.484E+01	3.426E+01	3.217E+01	3.125E+01	3.725E+01
	F-rank	1.180E+00	7.600E+00	5.320E+00	5.240E+00	8.720E+00	2.380E+00	4.080E+00	7.840E+00	7.040E+00	5.600E+00
	Time	2.246E+01	4.292E+01	3.747E+01	7.422E+01	8.901E+00	1.378E+01	1.384E+01	1.633E+01	1.051E+01	1.157E+01
F44	Best	2.213E+03	2.230E+03	2.235E+03	2.233E+03	2.262E+03	2.237E+03	2.228E+03	2.300E+03	2.302E+03	2.300E+03
	Ave	2.297E+03	2.300E+03	2.300E+03	2.303E+03	2.486E+03	2.300E+03	2.297E+03	2.301E+03	2.345E+03	2.324E+03
	Worst	2.301E+03	2.307E+03	2.305E+03	2.312E+03	3.861E+03	2.311E+03	2.314E+03	2.303E+03	2.323E+03	2.889E+03
	SD	1.754E+01	1.463E+01	1.358E+01	1.494E+01	4.725E+02	1.558E+01	3.281E+01	8.786E+01	1.850E+02	1.176E+02
	F-rank	2.320E+00	4.920E+00	5.400E+00	7.240E+00	9.360E+00	6.160E+00	6.320E+00	2.280E+00	7.880E+00	3.120E+00
	Time	2.335E+01	4.383E+01	3.847E+01	7.624E+01	9.843E+00	1.486E+01	1.496E+01	1.753E+01	1.153E+01	1.261E+01
F45	Best	2.600E+03	2.612E+03	2.608E+03	2.611E+03	2.613E+03	2.619E+03	2.609E+03	2.607E+03	2.603E+03	2.603E+03
	Ave	2.605E+03	2.626E+03	2.627E+03	2.620E+03	2.650E+03	2.629E+03	2.656E+03	2.617E+03	2.614E+03	2.614E+03
	Worst	2.609E+03	2.664E+03	2.657E+03	2.639E+03	2.691E+03	2.669E+03	2.703E+03	2.621E+03	2.637E+03	2.625E+03
	SD	2.183E+00	1.381E+01	1.336E+01	8.572E+00	2.001E+01	1.380E+01	2.278E+01	3.270E+00	9.340E+00	5.985E+00
	F-rank	1.040E+00	6.000E+00	5.960E+00	4.720E+00	8.720E+00	6.560E+00	9.280E+00	4.360E+00	4.760E+00	3.600E+00
	Time	2.453E+01	4.531E+01	3.976E+01	7.863E+01	1.061E+01	1.558E+01	1.568E+01	1.831E+01	1.224E+01	1.328E+01
F46	Best	2.400E+03	2.500E+03	2.500E+03	2.500E+03	2.509E+03	2.500E+03	2.413E+03	2.748E+03	2.727E+03	2.734E+03
	Ave	2.598E+03	2.734E+03	2.735E+03	2.736E+03	2.764E+03	2.570E+03	2.643E+03	2.753E+03	2.745E+03	2.746E+03
	Worst	2.739E+03	2.788E+03	2.783E+03	2.763E+03	2.844E+03	2.767E+03	2.806E+03	2.759E+03	2.782E+03	2.765E+03
	SD	1.232E+02	7.155E+01	7.161E+01	4.978E+01	7.500E+01	1.150E+02	1.417E+02	3.307E+00	1.380E+01	8.054E+00
	F-rank	1.940E+00	6.460E+00	6.520E+00	5.120E+00	8.680E+00	2.840E+00	5.640E+00	6.880E+00	5.400E+00	5.520E+00
	Time	2.489E+01	4.586E+01	4.002E+01	7.930E+01	1.112E+01	1.604E+01	1.620E+01	1.872E+01	1.275E+01	1.362E+01
F47	Best	2.898E+03	2.899E+03	2.898E+03	2.898E+03	2.904E+03	2.898E+03	2.6000E+03	2.898E+03	2.898E+03	2.898E+03
	Ave	2.909E+03	2.941E+03	2.933E+03	2.916E+03	2.953E+03	2.936E+03	2.916E+03	2.932E+03	2.929E+03	2.925E+03
	Worst	2.944E+03	2.969E+03	3.024E+03	2.948E+03	3.028E+03	2.953E+03	2.951E+03	2.946E+03	2.950E+03	2.951E+03
	SD	1.971E+01	1.661E+01	3.075E+01	2.307E+01	2.927E+01	2.079E+01	6.990E+01	2.169E+01	1.931E+01	2.364E+01
	F-rank	1.940E+00	6.600E+00	5.560E+00	4.640E+00	8.880E+00	6.320E+00	5.760E+00	5.020E+00	5.520E+00	4.760E+00
	Time	2.404E+01	4.513E+01	3.925E+01	7.772E+01	1.020E+01	1.516E+01	1.514E+01	1.775E+01	1.164E+01	1.274E+01
F48	Best	2.900E+03	2.800E+03	2.600E+03	2.600E+03	2.613E+03	2.6000E+03	2.600E+03	2.900E+03	2.832E+03	2.800E+03
	Ave	2.900E+03	3.139E+03	3.039E+03	3.035E+03	3.507E+03	2.949E+03	3.062E+03	2.932E+03	3.135E+03	2.965E+03
	Worst	2.900E+03	4.054E+03	4.121E+03	3.354E+03	4.616E+03	3.184E+03	3.662E+03	3.001E+03	3.927E+03	3.802E+03
	SD	9.282E-14	3.784E+02	3.192E+02	1.781E+02	5.491E+02	1.162E+02	2.081E+02	3.744E+01	3.825E+02	1.862E+02
	F-rank	2.140E+00	6.240E+00	5.320E+00	6.280E+00	8.640E+00	5.160E+00	6.680E+00	3.820E+00	6.280E+00	4.440E+00
	Time	2.532E+01	4.596E+01	4.049E+01	8.023E+01	1.141E+01	1.633E+01	1.632E+01	1.885E+01	1.296E+01	1.414E+01
F49	Best	3.087E+03	3.090E+03	3.090E+03	3.090E+03	3.095E+03	3.093E+03	3.100E+03	3.089E+03	3.092E+03	3.090E+03
	Ave	3.093E+03	3.100E+03	3.110E+03	3.095E+03	3.125E+03	3.108E+03	3.124E+03	3.090E+03	3.104E+03	3.093E+03
	Worst	3.097E+03	3.187E+03	3.164E+03	3.099E+03	3.226E+03	3.136E+03	3.181E+03	3.096E+03	3.153E+03	3.102E+03
	SD	2.265E+00	2.050E+01	2.479E+01	2.293E+00	3.888E+01	1.114E+01	1.992E+01	1.399E+00	1.548E+01	3.668E+00
	F-rank	3.160E+00	4.720E+00	6.400E+00	4.480E+00	8.480E+00	7.640E+00	9.040E+00	1.400E+00	6.120E+00	3.560E+00
	Time	2.595E+01	4.720E+01	4.187E+01	8.251E+01	1.182E+01	1.698E+01	1.693E+01	1.939E+01	1.337E+01	1.452E+01
F50	Best	2.800E+03	3.100E+03	3.100E+03	3.100E+03	3.170E+03	3.100E+03	3.100E+03	3.108E+03	3.168E+03	3.100E+03
	Ave	3.076E+03	3.354E+03	3.286E+03	3.336E+03	3.356E+03	3.161E+03	3.329E+03	3.313E+03	3.359E+03	3.317E+03
	Worst	3.100E+03	3.412E+03	3.732E+03	3.412E+03	3.732E+03	3.413E+03	3.504E+03	3.412E+03	3.459E+03	3.412E+03
	SD	8.307E+01	1.156E+02	1.722E+02	1.268E+02	1.190E+02	1.118E+02	1.534E+02	1.158E+02	9.553E+01	1.384E+02
	F-rank	1.000E+00	5.880E+00	6.460E+00	7.120E+00	7.080E+00	3.360E+00	7.360E+00	5.840E+00	7.320E+00	5.400E+00
	Time	2.930E+01	5.419E+01	4.611E+01	9.253E+01	1.270E+01	1.892E+01	1.911E+01	2.240E+01	1.499E+01	1.561E+01
F51	Best	3.134E+03	3.143E+03	3.141E+03	3.141E+03	3.214E+03	3.149E+03	3.160E+03	3.146E+03	3.141E+03	3.137E+03
	Ave	3.151E+03	3.225E+03	3.241E+03	3.196E+03	3.357E+03	3.191E+03	3.235E+03	3.162E+03	3.204E+03	3.181E+03
	Worst	3.178E+03	3.371E+03	3.430E+03	3.281E+03	3.528E+03	3.284E+03	3.368E+03	3.193E+03	3.326E+03	3.284E+03
	SD	1.004E+01	6.800E+01	8.133E+01	3.104E+01	9.956E+01	3.954E+01	6.011E+01	1.247E+01	5.265E+01	3.779E+01
	F-rank	1.680E+00	6.120E+00	6.760E+00	5.680E+00	9.440E+00	5.000E+00	7.000E+00	3.200E+00	5.760E+00	4.360E+00
	Time	3.195E+01	6.065E+01	5.342E+01	1.062E+02	1.450E+01	2.080E+01	2.082E+01	2.453E+01	1.665E+01	1.842E+01
F52	Best	3.581E+03	3.474E+03	3.529E+03	3.756E+03	3.162E+04	3.570E+03	3.100E+04	3.780E+03	3.112E+04	3.936E+03
	Ave	3.590E+03	5.507E+05	3.016E+05	1.599E+05	5.718E+04	6.381E+04	7.285E+05	1.703E+05	7.370E+05	1.555E+05
	Worst	5.669E+03	4.603E+06	1.252E+06	6.231E+05	2.425E+06	9.755E+05	4.030E+06	8.206E+05	1.754E+06	1.252E+06
	SD	5.759E+02	9.841E+05	5.006E+05	2.050E+05	5.525E+05	2.156E+05	1.179E+06	3.319E+05	7.027E+05	3.528E+05
	F-rank	2.200E+00	4.580E+00	4.120E+00	6.640E+00	7.880E+00	3.280E+00	7.480E+00	4.780E+00	8.440E+00	5.600E+00
	Time	4.455E+01	7.747E+01	6.590E+01	1.294E+02	2.553E+01	3.259E+01	3.387E+01	3.820E+01	2.885E+01	3.089E+01

Bold values indicate the best outcomes.

to prevent getting stuck in local minima. During the latter half, the diversity is rapidly decreased to guide the convergence speed towards the most promising region that has been identified thus far.

This section highlights the convergence behavior of MSA through the convergence curves displayed in the second column. The curves demonstrate that MSA has an accelerating reduction pattern during the optimization process, especially in the latter half. During this period, the population diversity is significantly reduced and the exploratory process transforms into an exploitation phase, speeding up the convergence toward the best-so-far solution. These results show that MSA has an appropriate convergence behavior, starting slowly to thoroughly search the space and avoid getting stuck in local minima, and then increasing the convergence speed in the final stages by shifting from exploration to exploitation towards the most promising area discovered thus far.

The average fitness history curve shown in the figure indicates the decline in performance achieved by MSA, demonstrating the competition among solutions to reach the near-optimal solution for the test functions. From the trajectory curve in Fig. 10, it can be seen that MSA gradually explores all the possible areas in the search space to find the best position in this dimension. As the number of function evaluations increases, the exploration process transforms into exploitation, focusing the search process on one aspect in the hope of finding a better solution. The

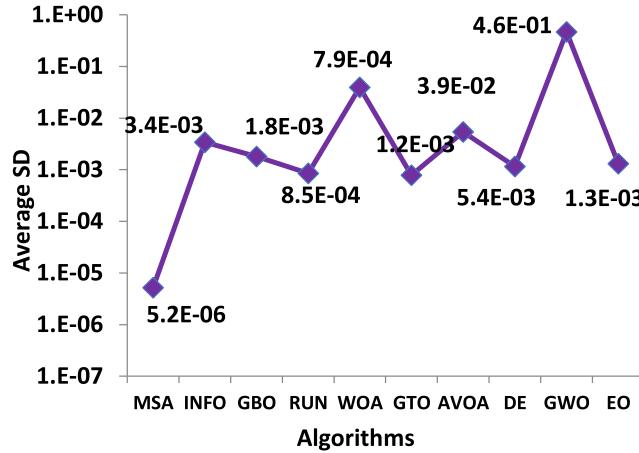


Fig. 9. Average SD on all selected CEC-2017 functions.

Table 8Findings (*p*-values) of the Wilcoxon rank-sum test on the CEC-2017 test suite (F24–F52).

Function	INFO	GBO	RUN	WOA	GTO	AVOA	DE	GWO	EO
F24	9.8988E-10	9.8988E-10	9.8988E-10	9.8988E-10	9.8988E-10	2.8050E-04	9.8988E-10	9.8988E-10	9.8988E-10
F25	3.1826E-09	8.8558E-10	2.4606E-10	2.4606E-10	1.1497E-09	2.4606E-10	2.6634E-05	2.4606E-10	2.4473E-10
F26	1.5967E-09	5.8547E-09	1.4157E-09						
F27	3.9010E-09	1.4160E-09	1.2557E-09	1.2557E-09	1.8004E-09	1.2557E-09	1.8036E-09	8.3354E-09	1.9426E-08
F28	3.8898E-10	3.8898E-10	3.8898E-10	3.8898E-10	3.8898E-10	3.8898E-10	2.0744E-02	3.8898E-10	8.6471E-10
F29	1.5967E-09	1.4157E-09	1.4157E-09	1.4157E-09	1.5967E-09	1.4157E-09	1.5967E-09	5.8547E-09	3.2057E-08
F30	1.3823E-09	2.8108E-09	1.3047E-09	1.3047E-09	1.3035E-09	1.3047E-09	1.3047E-09	2.1109E-09	2.2286E-08
F31	5.9677E-10	1.3684E-10	1.3762E-10	1.3762E-10	1.3762E-10	1.3762E-10	3.3706E-01	1.3762E-10	2.9681E-09
F32	8.2805E-09	2.8980E-09	1.3090E-07	1.5967E-09	4.1259E-09	9.2880E-09	1.3090E-07	1.0414E-08	8.5468E-08
F33	6.4605E-09	8.1390E-09	1.3890E-09	1.3890E-09	4.5563E-09	1.3890E-09	9.6903E-01	1.3890E-09	1.4769E-06
F34	3.8987E-05	1.8356E-08	1.4157E-09	1.4157E-09	4.1259E-09	1.4157E-09	3.0175E-07	1.4157E-09	4.1259E-09
F35	2.2857E-09	1.4157E-09	1.4157E-09	1.4157E-09	1.4157E-09	1.4157E-09	1.7007E-02	1.4157E-09	1.4157E-09
F36	3.3254E-09	1.1266E-09	1.1266E-09	1.1266E-09	1.1266E-09	1.1266E-09	8.5672E-02	1.1266E-09	1.1266E-09
F37	2.0536E-08	1.4157E-09	1.4157E-09	1.4157E-09	1.4157E-09	1.4157E-09	1.3673E-03	1.4157E-09	1.4157E-09
F38	1.4157E-09	1.5967E-09	1.4157E-09	1.4157E-09	2.5742E-09	1.4157E-09	1.5633E-03	1.4157E-09	1.4157E-09
F39	1.4144E-09	1.4144E-09	1.4144E-09	1.4144E-09	1.4144E-09	1.4144E-09	2.0723E-02	1.4144E-09	1.4144E-09
F40	1.4157E-09	1.4157E-09	1.4157E-09	1.4157E-09	1.4157E-09	1.4157E-09	3.8259E-01	1.4157E-09	1.4157E-09
F41	1.4080E-09	1.4080E-09	1.4080E-09	1.4080E-09	1.4080E-09	1.4080E-09	1.0943E-05	1.4080E-09	1.4080E-09
F42	6.4389E-10	6.8621E-10	6.4515E-10	6.4515E-10	6.4515E-10	6.4515E-10	2.7835E-03	6.4515E-10	6.8554E-10
F43	2.5605E-08	2.1100E-06	7.6346E-06	6.2516E-09	1.1984E-03	1.5174E-04	1.8996E-09	3.4623E-09	3.4786E-08
F44	4.9626E-07	8.4862E-08	1.4029E-09	1.4029E-09	1.4029E-09	3.1960E-06	3.8980E-02	5.8060E-09	9.0444E-03
F45	5.2121E-09	1.8002E-09	2.2857E-09	1.4157E-09	4.6381E-09	2.5677E-08	4.4598E-08	3.5800E-08	2.8695E-08
F46	1.0951E-08	1.9745E-09	6.4331E-09	1.0486E-07	2.5037E-01	2.4248E-01	1.0951E-08	6.1121E-08	1.4386E-08
F47	2.5991E-07	1.4620E-05	2.1133E-06	6.5506E-08	7.1931E-04	1.5975E-05	1.9035E-05	1.0248E-05	7.1590E-04
F48	1.7786E-02	7.4145E-09	7.0486E-04	3.1460E-10	1.0179E-07	3.6875E-03	1.7792E-02	6.3430E-09	9.5707E-08
F49	2.3031E-03	4.1019E-04	2.2129E-01	1.9844E-09	1.7652E-07	1.3840E-09	3.9513E-04	1.0981E-02	5.2173E-01
F50	1.2008E-06	6.2524E-08	9.0155E-09	6.3164E-09	1.1156E-06	4.0465E-08	1.1462E-07	1.8176E-08	2.6998E-08
F51	2.8695E-08	3.2614E-09	1.3079E-08	1.4157E-09	2.7218E-07	1.8002E-09	3.7045E-07	1.8356E-08	9.6957E-06
F52	3.6117E-02	5.2048E-03	1.4157E-09	4.6381E-09	4.1349E-04	1.4157E-09	8.5407E-08	1.4157E-09	2.2837E-09

trajectory curve suggests that MSA starts with an exploratory trend and then switches to exploitation. Finally, the last column in Fig. 10 displays the history of positions obtained by MSA. This column shows that MSA does not follow the same pattern for different test functions. For example, for F24, MSA explores a large portion of the search space, then focuses its search within the interval $[-100, 0]$ to exploit this region and search for the near-optimal solution. The search history graph shows that MSA's performance for composition and multimodal test functions is more dispersed, while its performance for unimodal test functions is more focused around the optimal points.

4.6. Evaluation of the convergence ability

In this section, the convergence curve obtained by MSA will be depicted with that obtained by the rival algorithms in Fig. 11 to show their characteristics. The figure depicts that MSA's convergence curves have a pattern of rapid decrease, particularly in the latter half of the optimization process. The specific performance comparison between MSA and rival algorithms, as shown in the convergence curves in Fig. 11, is described below:

- The WOA has a poor convergence rate for three families (multimodal, composition, and hybrid) of the test functions compared to all optimizers, followed by the GWO.

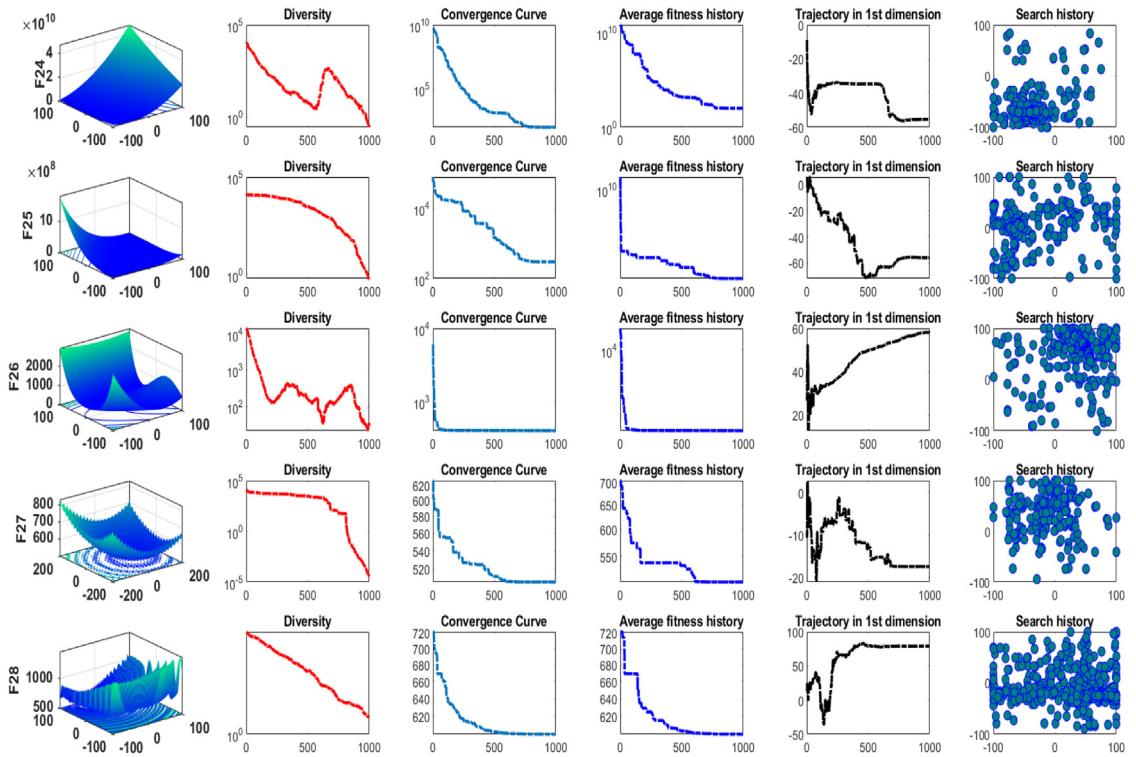


Fig. 10. Diversity, convergence curve, average fitness history, trajectory and search history.

- The other rival optimizers are competitive since one of them has a strong convergence rate on a test function, and a poor another.
- The MSA optimizer outperforms all its rival algorithms for unimodal, multimodal, hybrid, and composition test functions. This is due to its ability to effectively balance exploration and exploitation, which allows it to avoid getting stuck in local minima while also speeding up the convergence rate towards the most promising solution found so far.
- The convergence curves demonstrate that MSA excels in balancing both exploration and exploitation operations, leading to two key advantages: (1) preventing stagnation in local minima and (2) quickly converging towards the best solution found so far.

4.7. MSA's scalability analysis

This section assesses the MSA's performance over the high-dimensional mathematical test functions and the outcomes obtained by the proposed are compared with those of the rival optimization algorithms: INFO, GBO, RUN, WOA, GTO, AVOA, DE, GWO, and EO. Eight multimodal and composition test functions: F27, F28, F29, F30, F31, F35, F36, and F38 are selected for conducting the scalability analysis of the proposed algorithms. The CEC-2017 test suite is only proposed for dimensions 100, 50, 30, and 10. Therefore, the optimizers' performances are only analyzed according to these dimensions in this study. The maximum number of function evaluations and population size are considered as used in the previous experiments for each algorithm.

Fig. 12 displays the average fitness values of each algorithm on eight selected test functions in the considered dimensions, which demonstrates that MSA has a superior performance for these functions over various validated dimensions: 10, 30, 50, and 100. This section affirms that MSA has a strong performance for both small-scale and high-scale optimization problems and subsequently it is a strong alternative to all the existing optimizers for tackling any optimization problem over any number of dimensions.

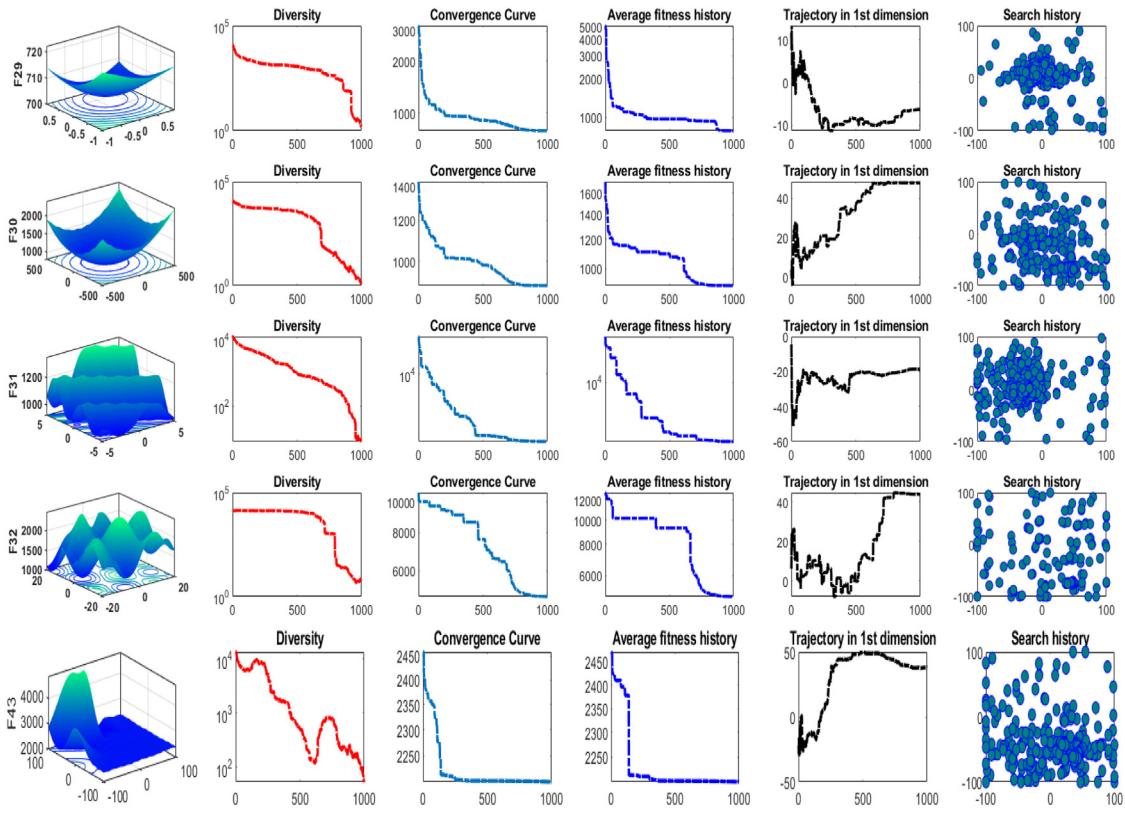


Fig. 10. (continued).

Table 9

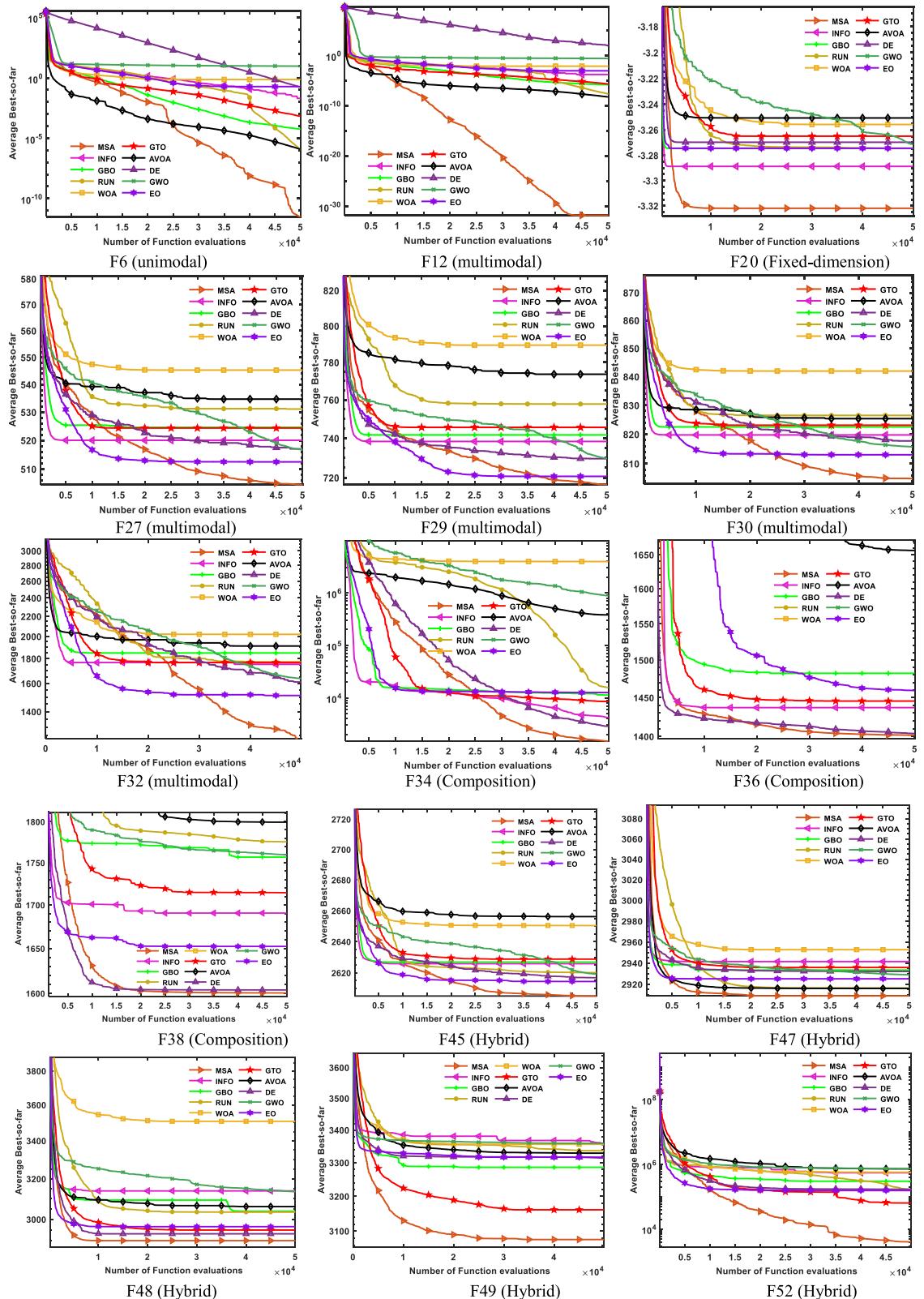
The fittest values obtained by MSA with maximizing the function evaluations.

Function	opt. Value	N	NFEs									
	Dim=10											
F24	100.0000	30	171013	100.0000	30	1189230	100.0000	30	2279400	100.0000	40	7279306
F25	300.0000	30	92340	300.0000	30	868760	300.0000	30	2627220	300.0000	30	10065300
F26	400.0000	30	316026	400.0000	30	1246480	400.0000	30	3013733	403.0000	40	—
F27	500.0000	30	3417200	530.836	150	—	591.5359	50	—	762.6683	40	—
F28	600.0000	30	71920	600.0000	30	223147	600.0000	80	790693	600.0000	150	2537650
F29	710.4613	50	—	759.2161	50	—	840.3480	50	—	1183.6517	50	—
F30	800.0000	100	2493233	835.8184	100	—	894.3897	100	—	1031.8249	100	—
F31	900.0000	30	35370	900.0000	30	108090	900.0000	100	664433	900.9086	200	—

Bold values indicate the optimal outcomes.

4.8. MSA's performance analysis over higher function evaluations

This section investigates how the maximum number of function evaluations and population size influences MSA's ability to reach a near-optimal point. Within this experiment, eight test functions are selected to investigate the performance of MSA for reaching their optimal values in the dimension of 10, 30, 50, and 100. The obtained outcomes by MSA on those functions are displayed in Table 9, which includes the optimal value (opt. Value), population size (N), and the maximum number of function evaluations (NFEs). By inspecting this table, it can be observed that the proposed could reach the optimal values for seven test functions under a dimension of 10. For dimensions 30 and 50, it could reach the optimal value for 5 out of 8 investigated test functions; for dimension 100, it could find the optimal values for 3 test functions and the values for the other functions are so near to the optimal value.

**Fig. 11.** Averaged convergence curve of various optimizers on some test functions.

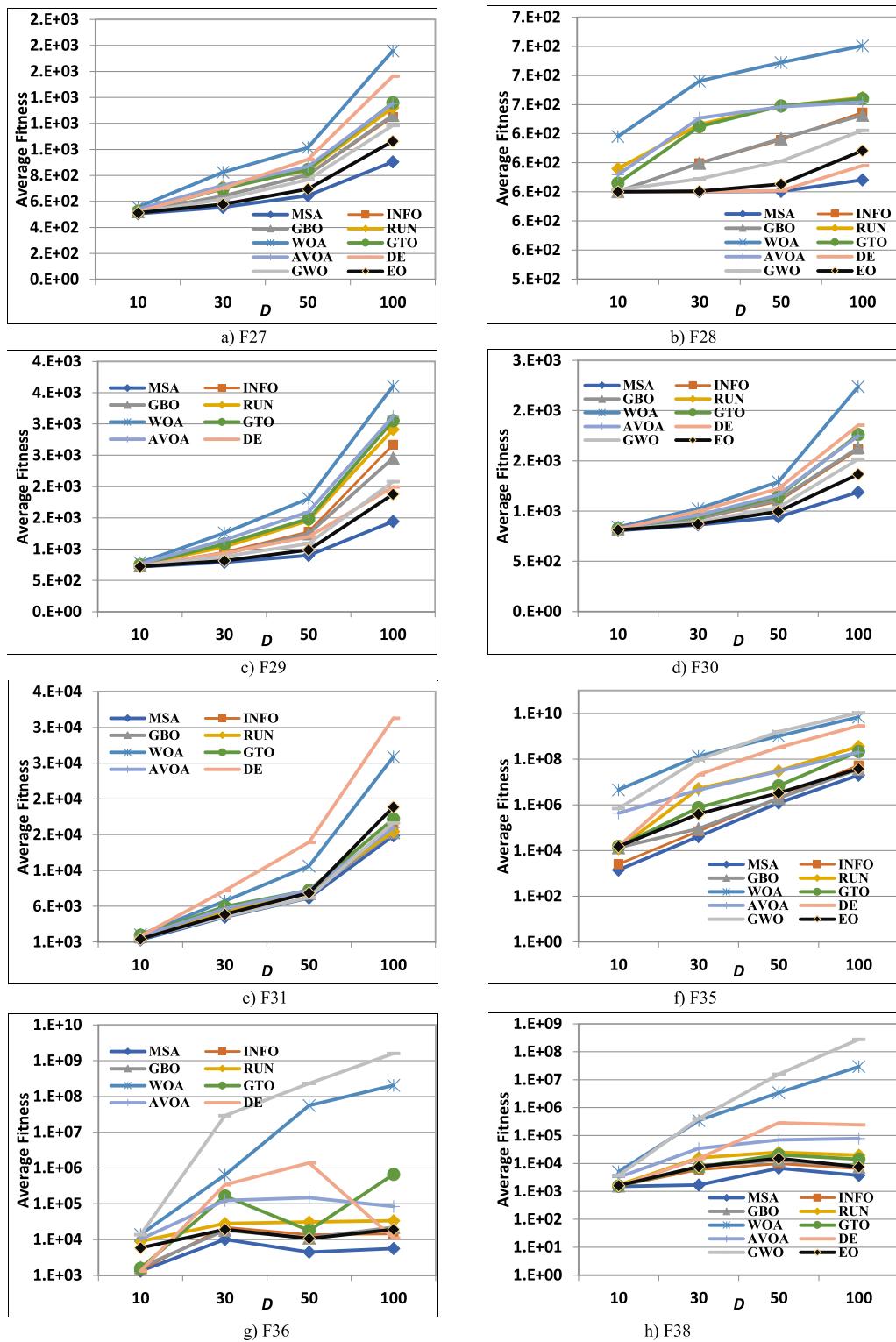


Fig. 12. Scalability analysis of MSA.

5. Application I: MSA's performance on engineering design optimization problems

This section covers the experiments carried out to evaluate the performance of MSA on five real-world engineering problems that have constraints: a welded beam, a tension/compression spring, a pressure vessel, a 3-bar truss, and a 10-bar truss. Several penalty methods are available to address the constraints of these problems, including static, dynamic, annealing, adaptive, co-evolutionary, and death penalty [120]. In this study, the death penalty method is employed along with MSA and other competing optimizers to neglect infeasible solutions during the optimization process by assigning them a large objective value in the case of minimization.

5.1. The welded beam design problem

In this study, the performance of MSA is evaluated using the Welded Beam Design (WBD) problem. More descriptions for this problem in addition to its structure are available in [61] The goal is to find the optimal values for four design variables (h , l , t , and b) that minimize the total fabrication cost of a welded beam while satisfying seven constraints. These constraints include tip deflection, shear stress, weld coverage, bending stress, buckling load, weld thickness, and cost. The mathematical representation of the WBD problem is given as follows:

Solution representation as that

$$X = [x_1 \ x_2 \ x_3 \ x_4] = [h \ l \ t \ b]$$

- The objective value is calculated utilizing the following equation:

$$f(X) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)$$

- The constraints of this problem are mathematically formulated:

$$g_1(X) = \tau(X) - \tau_{max} \leq 0$$

$$g_2(X) = \sigma(X) - \sigma_{max} \leq 0$$

$$g_3(X) = \delta(X) - \delta_{max} \leq 0$$

$$g_4(X) = x_1 - x_4 \leq 0$$

$$g_5(X) = P - P_c(X) \leq 0$$

$$g_6(X) = 0.125 - x_1 \leq 0$$

$$g_7(X) = 1.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \leq 0$$

Where

$$\begin{aligned} \tau(X) &= \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2} \\ \tau' &= \frac{P}{\sqrt{2}x_1x_2}, \quad \tau'' = \frac{MR}{J}, \quad M = P\left(L + \frac{x_2}{2}\right) \end{aligned}$$

$$\begin{aligned} R &= \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2} \\ J &= 2 \left\{ \sqrt{2}x_1x_2 \left[\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2 \right] \right\} \end{aligned}$$

$$\sigma(X) = \frac{6PL}{x_4x_3^2}, \quad \delta(X) = \frac{6PL^3}{Ex_3^2x_4}$$

$$P_c(X) = \frac{4.0134E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2} \left(1 - \frac{x_3}{2L} \sqrt{\frac{E}{4G}} \right)$$

$$P = 6000, L = 14, E = 30 \times 10^6, G = 12 \times 10^6,$$

$$\tau_{max} = 13600, \sigma_{max} = 30000, \delta_{max} = 0.25$$

Table 10

Comparison on the Welded beam design problem.

Algorithms	x_1	x_2	x_3	x_4	Best	Ave	Worst	SD	F-rank	Time
MSA	0.20572	3.47070	9.03662	0.20573	1.7248658	1.7248658	1.7248658	2.4E-16	1.6	23.4
INFO	0.20572	3.47070	9.03662	0.20573	1.7248658	1.7248659	1.7248659	5.3E-09	3.1	39.6
GBO	0.20572	3.47070	9.03662	0.20573	1.7248658	1.7248659	1.7248672	2.7E-07	4.0	31.2
RUN	0.20572	3.47070	9.03664	0.20573	1.7248673	1.7566028	1.9218500	4.2E-02	8.1	66.0
WOA	0.20297	3.92134	9.09540	0.20544	1.8111932	2.4642498	4.4797213	7.2E-01	9.9	8.6
GTO	0.20572	3.47070	9.03662	0.20573	1.7248658	1.7483480	2.2172739	9.9E-02	6.0	13.7
AVOA	0.20572	3.47076	9.03663	0.20573	1.7258556	1.7444667	1.8713394	3.2E-02	8.2	13.7
DE	0.20572	3.47070	9.03662	0.20573	1.7248658	1.7248658	1.7248658	2.6E-16	1.6	16.3
GWO	0.20556	3.47457	9.03689	0.20574	1.7253983	1.7265616	1.7344144	1.9E-03	7.3	9.4
EO	0.20572	3.47070	9.03662	0.20573	1.7248658	1.7249802	1.7257613	2.5E-04	5.1	11.6

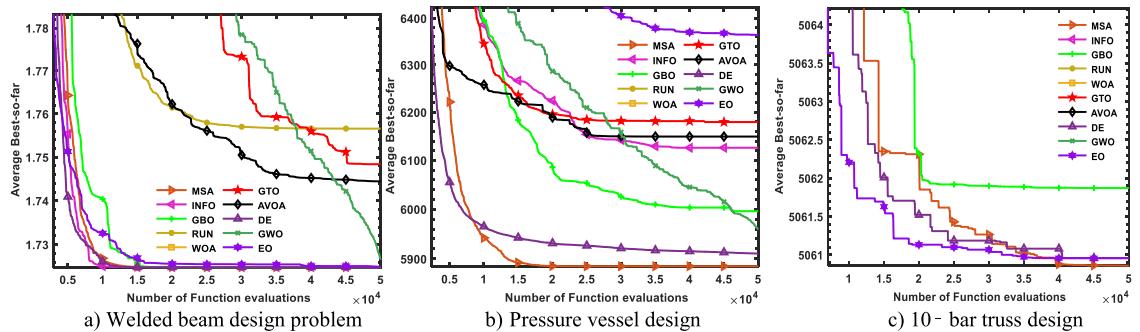
**Fig. 13.** Averaged convergence curve of various optimizers on some investigated real-world problems.

Table 10 presents the results obtained from applying MSA and nine other optimization algorithms to the WBD problem. It shows that MSA is competitive in terms of its fitness values and outperforms two highly-performing optimizers, INFO, and DE, in terms of SD. Additionally, Fig. 13(a) illustrates the convergence speed achieved by each algorithm on the WBD problem, revealing that MSA has a faster convergence speed than all other optimizers.

5.2. Tension/compression spring design problem

The objective of this problem is to minimize the weight of the tension/compression spring [122–124]. This problem requires satisfying constraints such as shear stress, surge frequency, and deflection in the optimum design. To achieve this, three design variables, mean coil diameter (D), wire diameter (d), and the number of active coils (N), need to be accurately determined. The mathematical model for this problem is outlined as follows:

Consider $X = [x_1 \ x_2 \ x_3] = [d \ D \ N]$

$$\text{Minimize } f(X) = (x_3 + 2)x_2x_1^2$$

$$\text{Subject to } g_1(X) = 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0$$

$$g_2(X) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} \leq 0$$

$$g_3(X) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0$$

$$g_4(X) = \frac{x_1 + x_2}{1.5} - 1 \leq 0$$

Variable range $0.05 \leq x_1 \leq 2.00$

Table 11

Comparison of the Tension/compression spring design problem.

Algorithms	x_1	x_2	x_3	Best	Ave	Worst	SD	F-rank	Time
MSA	0.05168481212	0.35661552912	11.29496069555	0.012665	0.0126668	0.0126789	2.8E-06	1.4	22.5
INFO	0.05087580468	0.33746619698	12.51376624017	0.012677	8.0123759	100.0025000	2.8E+01	6.6	36.2
GBO	0.05174042922	0.35795468688	11.21682132529	0.012665	0.0126775	0.0127264	1.7E-05	2.9	30.8
RUN	0.05000132958	0.31745506525	14.02533862292	0.012719	0.0150712	0.0177733	1.7E-03	8.3	64.9
WOA	0.05217022463	0.36840401456	10.63534456369	0.012669	0.0146865	0.0177742	1.5E-03	8.0	8.0
GTO	0.05135001731	0.34861588489	11.78025159217	0.012667	0.0128248	0.0132721	1.7E-04	5.2	12.7
AVOA	0.05259999906	0.37903075333	10.09144376514	0.012680	0.0130665	0.0140543	3.4E-04	6.5	12.9
DE	0.05168906164	0.35671775322	11.28896496508	0.012665	16.0111195	100.0025000	3.7E+01	3.5	15.2
GWO	0.05204238711	0.36523546373	10.80810541658	0.012670	16.0110783	100.0025000	3.7E+01	5.1	8.4
EO	0.05184072025	0.36037636112	11.07768559141	0.012666	40.0086782	100.0025000	5.0E+01	7.4	10.0

$$0.25 \leq x_2 \leq 1.30$$

$$2.00 \leq x_3 \leq 15.0$$

The results of the near-optimal values of three design variables, corresponding weight, and statistical information such as Ave, Worst, and SD are presented in [Table 11](#). These results were obtained through the MSA and nine other rival algorithms. The table demonstrates that MSA outperforms the other algorithms as it reaches better Ave, Worst, and SD values.

5.3. Pressure vessel design problem

The goal of this problem is to minimize total production cost by determining the optimal values of four design variables while satisfying pressure requirements as the optimization constraint. These design variables include the inner radius (R), the thickness of the head (Th), the thickness of the shell (Ts), and the length of the cylindrical portion (L). The structure of the pressure vessel in addition to more descriptions are presented in [\[62\]](#). This optimization problem is modeled as follows [\[58\]](#):

$$\text{Consider } X = [x_1 \ x_2 \ x_3 \ x_4] = [Ts \ Th \ R \ L]$$

$$\text{Minimize } f(X) = 0.6224x_1x_3x_4 + 1.7781x_3x_1^2 + 3.1661x_4x_1^2 + 19.84x_3x_1^2$$

$$\text{Subject to } g_1(X) = -x_1 + 0.0193x_3 \leq 0$$

$$g_2(X) = -x_3 + 0.00954x_3 \leq 0$$

$$g_3(X) = -\pi x_4x_3^2 - \frac{4}{3}x_3^3 + 1296000 \leq 0$$

$$g_4(X) = x_4 - 240 \leq 0$$

$$\text{Variable range } 0 \leq x_1 \leq 99$$

$$0 \leq x_2 \leq 99$$

$$10 \leq x_3 \leq 200$$

$$10 \leq x_4 \leq 200$$

[Table 12](#) displays the results of the pressure vessel design problem solved by MSA and nine other optimization algorithms, indicating that MSA outperforms all other algorithms in terms of average, worst, and standard deviation values. Moreover, MSA's performance is comparable with that of INFO, DE, and GTO for the best value. Furthermore, [Fig. 13\(b\)](#) illustrates the average convergence speed of each algorithm, showing that MSA is faster than all other algorithms.

5.4. The three-bar truss design problem

The goal of this problem is to minimize the weight of the three-bar truss by determining the optimal values of three variables, namely the areas of bars 1, 2, and 3, while satisfying specific constraints. More details about this

Table 12

Comparison of the Pressure Vessel Design problem.

Algorithms	x_1	x_2	x_3	x_4	Best	Ave	Worst	SD	F-rank	Time
MSA	0.77818	0.38466	40.31962	200.00000	5885.4341746	5885.4341746	5885.4341746	1.4E-10	1.6	43.5
INFO	0.77818	0.38466	40.31962	200.00000	5885.4341746	6126.0132060	6756.6619389	2.2E+02	6.2	69.5
GBO	0.77818	0.38466	40.31962	199.99999	5885.4347683	5995.9967571	6651.0611895	2.1E+02	4.2	57.9
RUN	0.77824	0.38479	40.32279	199.95652	5885.5998254	6764.7255698	7319.2854820	6.7E+02	7.5	118.9
WOA	0.79425	0.42755	40.59944	196.14104	6270.7118691	7420.6512865	8414.7105871	6.4E+02	9.4	19.6
GTO	<u>0.77818</u>	<u>0.38466</u>	<u>40.31962</u>	<u>200.00000</u>	5921.1706337	6179.4803546	6755.5225366	2.3E+02	6.7	25.6
AVOA	0.77839	0.38476	40.33069	199.84590	5910.1147938	6149.2270959	6719.6846538	2.4E+02	6.5	25.9
DE	0.77818	0.38466	40.31962	200.00000	5885.4341746	5910.6130901	6078.8999171	5.3E+01	2.4	32.8
GWO	0.77880	0.38556	40.33991	199.74704	5887.5662022	5958.8271660	7275.2219087	2.8E+02	4.2	20.2
EO	0.78751	0.38927	40.80326	193.37513	5885.4341746	6364.3674869	7319.1137859	5.3E+02	6.4	22.2

Table 13

Comparison of Three-bar truss design problem.

Algorithms	x_1	x_2	Best	Ave	Worst	SD	F-rank	Time
MSA	0.788675	0.408248	263.895843	263.89584	263.89584	3.3E-14	1.5	17.4
INFO	0.788675	0.408248	263.895843	263.89584	263.89585	4.4E-07	3.7	29.2
GBO	0.788672	0.408258	263.895843	263.89584	263.89584	2.9E-08	3.9	25.5
RUN	0.788663	0.408282	263.895850	263.91616	263.98553	3.0E-02	8.4	53.4
WOA	0.790759	0.402386	263.899011	264.69461	268.39339	1.3E+00	9.8	2.6
GTO	0.788675	0.408249	263.895843	263.89584	263.89585	1.0E-06	4.4	7.8
AVOA	0.788689	0.408209	263.895844	263.89881	263.91287	4.1E-03	7.6	8.1
DE	0.788675	0.408248	263.895843	263.89584	263.89584	2.8E-14	1.5	9.4
GWO	0.788675	0.408248	263.895843	263.89762	263.90547	2.1E-03	7.9	2.9
EO	0.788675	0.408248	263.895843	263.89592	263.89635	1.1E-04	6.2	5.3

problem in addition to its structure are presented in [75,125]. This problem is mathematically expressed as follows:

Considers $X = [x_1 \ x_2]$

$$\text{Minimize } f(X) = (2\sqrt{2}x_1 + x_2) \times l$$

$$\text{Subject to } g_1(X) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0$$

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

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

Variable range $0 \leq x_1, x_2 \leq 1$

$$l = 100 \text{ cm}, P = 2 \frac{\text{KN}}{\text{cm}^2}, \sigma = 2 \frac{\text{KN}}{\text{cm}^2}$$

The results obtained by MSA when solving this problem are compared with those of INFO, GBO, RUN, WOA, GTO, AVOA, DE, GWO, and EO (See Table 13). According to this table, MSA could be competitive with DE, GBO, and INFO, and superior to the others for the Ave, Worst, and SD of the fitness values obtained within 30 independent runs. Additionally, the optimal values of the design variables and corresponding fitness values obtained by each algorithm are also stated in Table 13.

5.5. The 10-bar truss design problem

The final engineering design challenge addressed in this study is the 10-bar truss design problem which involves optimizing 10 design variables to reach the optimal design while satisfying 10 stress constraints and 12 displacement constraints. Further information regarding this problem can be found in [122].

Table 14

Comparison of the 10-bar truss design problem.

Algorithms	MSA	INFO	GBO	RUN	WOA	GTO	AVOA	DE	GWO	EO
x_1	30.506984	30.575667	30.586281	30.513710	34.932936	30.651351	31.115350	30.769023	30.950475	30.737004
x_2	0.100000	0.100000	0.100000	0.100012	0.113743	0.100000	0.100000	0.100000	0.114518	0.100000
x_3	23.178171	23.192306	23.152780	23.625674	22.103159	22.692440	23.245239	23.075296	23.104066	23.177082
x_4	15.249633	15.175915	15.249114	15.200738	14.250425	15.264766	14.931481	15.191436	15.272088	15.279246
x_5	0.100000	0.100000	0.100000	0.100000	0.100910	0.100000	0.100000	0.100056	0.100044	0.100059
x_6	0.545710	0.569709	0.545932	0.436313	0.100910	0.564570	0.511706	0.546894	0.348177	0.561942
x_7	21.051289	21.154244	21.165470	21.019808	18.930383	21.169215	21.191234	21.087841	21.091466	20.885203
x_8	7.460120	7.475991	7.480879	7.424337	8.226653	7.519120	7.501183	7.486320	7.492405	7.447889
x_9	0.100000	0.100002	0.100000	0.100005	0.190320	0.100000	0.100000	0.100000	0.103722	0.100002
x_{10}	21.521741	21.381911	21.351238	21.404581	23.582377	21.572061	21.132784	21.390893	21.435863	21.508313
Best	5060.86124	5060.98353	5060.98056	5062.12636	5170.59370	5061.36536	5061.91226	5061.07381	5067.88697	5060.99333
Ave	5060.88925	5067.42910	5068.83625	5076.15320	6549.90718	5172.28490	5081.09647	5060.93743	5077.24800	5068.25121
Worst	5060.97495	5078.20875	5085.85947	5081.93522	8807.15157	5551.17886	5106.18972	5061.06274	5091.56653	5078.03543
SD	2.6E-02	7.1E+00	8.3E+00	5.3E+00	9.1E+02	1.2E+02	8.8E+00	4.1E-02	8.5E+00	8.0E+00
F-rank	1.1	4.3	4.9	6.5	10.0	8.2	7.2	1.9	6.5	4.4
Time	430.2	459.9	441.7	888.5	395.9	402.7	408.5	414.4	396.6	400.2

The performance of MSA in comparison to other optimization algorithms, including INFO, GBO, RUN, WOA, GTO, AVOA, DE, GWO, and EO, is shown in [Table 14](#). This table displays the optimized design variables, as well as the best, average, worst, and standard deviation of the fitness values obtained by each algorithm over 30 independent runs. The results indicate that MSA outperforms all other algorithms. To assess the performance of the proposed algorithm, its convergence speed was also analyzed. [Fig. 13\(c\)](#) presents the average fitness values obtained by each algorithm during the optimization process; this figure appears the superiority of MSA over the other optimizers.

6. Application II: Parameter estimation of photovoltaic modules

In this section, MSA will be applied to estimate the unknown parameters of the photovoltaic modules to further show its effectiveness for solving real-world problem. Those modules might be based on the single-diode model (SDM), double-diode model (DDM), or triple-diode model (TDM). The SDM has five unknown parameters that are not given in the manufactured sheet, the DDM involves seven unknown parameters, and the TDM contains nine unknown parameters. Both SDM and DDM are considered in this study for estimating their unknown parameters using the proposed MSA. In this study, a solar cell known as RTC France cell, and a PV module called Photowatt-PWP201 (PWP) [126] are used to assess the MSA's performance. They are used due to being widely used in the literature [127]. The properties of both the RTC France cell and PWP module at the standard conditions (STC) are the short-circuit current point (I_{SC}), the maximum output current (I_m), the open-circuit current–voltage (V_{oc}), the maximum output power (P_m), the maximum output voltage (V_m), the short-circuit current-temperature factor (k_i), the number of cells in PV modules (N_s), and the temperature coefficient of open-circuit voltage (k_v) and defined in [Table 15](#) [128]. The objective function of this problem is based on the root mean squared error (RMSE) between the measured and predicted current. The mathematical model for this function is defined as follows:

$$RMSE = f(\vec{x}_i^t) = \sqrt{\frac{1}{M} * \sum_{k=1}^M (I_m - I_e(V_e, X_i))^2}$$

where I_e is the estimated current, described in detail later, I_m is the measured current. M is the total number of measured data points. \vec{x}_i^t includes the parameters estimated by MSA at iteration t .

As defined before, the metaheuristics distribute their solutions within the upper value and lower value for each dimension. Therefore, to apply MSA to tackling the parameter estimation of PV models, we must define the search boundary for each unknown parameter. According to several studies in the literature, the search boundary for each parameter is as defined in [Table 16](#). To observe the effectiveness of MSA, it is compared to eight metaheuristics, including marine predators algorithm (MPA) [128], grey wolf optimizer (GWO), dandelion optimizer (DO) [129], Fick's law algorithm (FLA) [130], slime mould algorithm (SMA) [131], pelican optimization algorithm (POA) [132], Nutcracker optimization algorithm (NOA) [62], LSHADE_cnEpSin (cnEpSin) [133], and equilibrium optimizer

Table 15

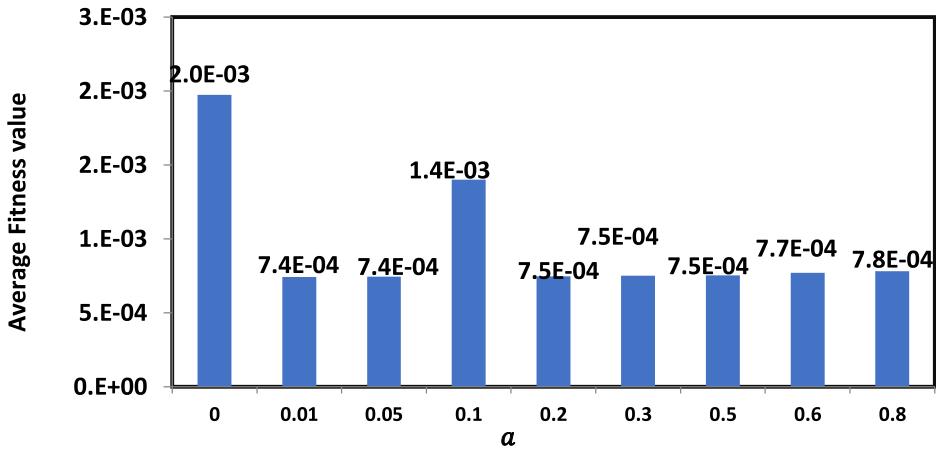
Properties of RTC France solar cell and PV models.

characteristics	RTC	PWP
P_m [W]	0.31	11.5
V_m [V]	0.459	12.649
I_m [A]	0.6755	0.912
V_{oc} [V]	0.5736	16.7785
I_{SC} [A]	0.7605	1.0317
N_s	1	36
K_i	0.000387	0.0008
K_v	-0.003739	-0.0725

Table 16

The search boundary of unknown parameters.

	I_{ph} (A)	I_{sdi} (A), $i \in 1:3$	R_s (Ω)	R_{sh} (Ω)	$a1$	$a2$	$a3$
U	$1.1I_{SC}$	$10 \mu\text{A}$	0.5	500	2	2	2
L	$0.9I_{SC}$	1 nA	0	0	1	1.2	1.4

**Fig. 14.** Tuning the parameter a .

(EO). All parameters of those algorithms with the exception of T and N , are set as defined in the cited paper, while Both T and N are set to 50000 and 25, respectively, to ensure a fair comparison. Regarding the controlling parameters of MSA, extensive experiments under several values for each parameter are conducted. From those experiments, we found that MSA could perform well under the same values reported in the sensitivity analysis section for all controlling parameters, with the exception of a , which could maximize the MSA's performance when it is set to 0.01, as shown in Fig. 14.

6.1. Single-diode model

As structured in Fig. 15, the SDM is comprised of a single diode and its output could be computed according to the following equation:

$$I = I_{ph} - I_D - I_{sh}$$

where I_{ph} is the photo-generated current [134] and I_D is the current of the diode current that could be estimated according to the following equation:

$$I_D = I_{sd}(\exp\left(\frac{V + I * R_s}{n * V_t}\right) - 1)$$

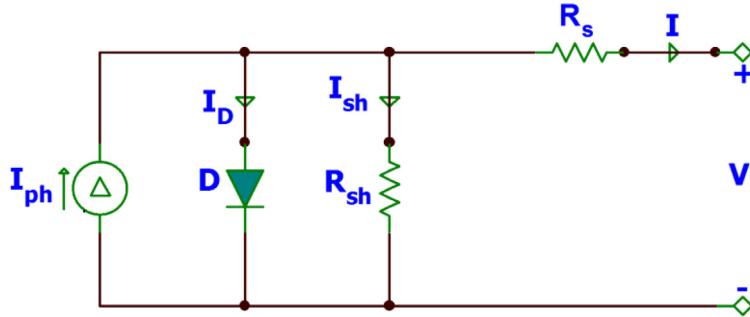


Fig. 15. SDM's equivalent circuit.

where I_{sd} represents the diode's reverse saturation current, V stands for the output voltage, R_s represents the series resistance, n represents the ideality factor of the diode, and V_t could be computed according to the following formula:

$$V_t = \frac{k * T}{q}$$

where T represents the junction temperature degree in kelvin, k involves a constant value of $1.3806503 \times 10^{-23}$ J/K to represent the Boltzmann constant, and q includes a constant value of $1.60217646 \times 10^{-19}$ C to represent the electron charge. I_{sh} could be computed using the following formula:

$$I_{sh} = \frac{V + I * R_s}{R_{sh}}$$

where R_{sh} is the shunt resistance. By Substitution, I could be extended as follows:

$$I = I_{ph} - I_{sd}(\exp\left(\frac{q * (V + I * R_s)}{n * k * T}\right) - 1) - \frac{V + I * R_s}{R_{sh}}$$

The mathematical model of SDM includes five unknown parameters (I_{ph} , I_{sd} , n , R_s , R_{sh}) that have to be accurately estimated to emulate the SDM accurately. To estimate them, MSA and some compared algorithms with the previously mentioned objective function are employed to determine the best algorithm which can estimate them with higher accuracy. Those algorithms are executed 30 independent times for both the RTC France cell and PWP module and the best, ave, and worst objective values, in addition to SD are computed and presented in Table 17. This table shows the superiority of MSA, where it could be the best for all the performance measures. To show the difference between the outcomes of MSA and those of the other algorithms, the p -value between MSA and each compared algorithm is computed and presented in the same table, which shows that MSA produces significantly-different outcomes. In addition, Fig. 16 shows the convergence curves of MSA and the other algorithms on both RTC France and the PWP module. Inspecting this figure shows that MSA has better convergence speed.

6.2. Double-diode model

Typically, the performance of SDM significantly deteriorated at low irradiance levels and is not a good alternative for several applications. Therefore, the DDM was developed to overcome the drawbacks of the SDM by employing an additional diode assisting in improving the performance in several case [135]. The DDM is compounded of two diodes, as depicted in Fig. 17. The first is a rectifier, and the second is added to void the current of the recombination and the non-idealities at the SC. The output of the DDM could be calculated according to the following formula:

$$I = I_{ph} - I_{D1} - I_{D2} - I_{sh}$$

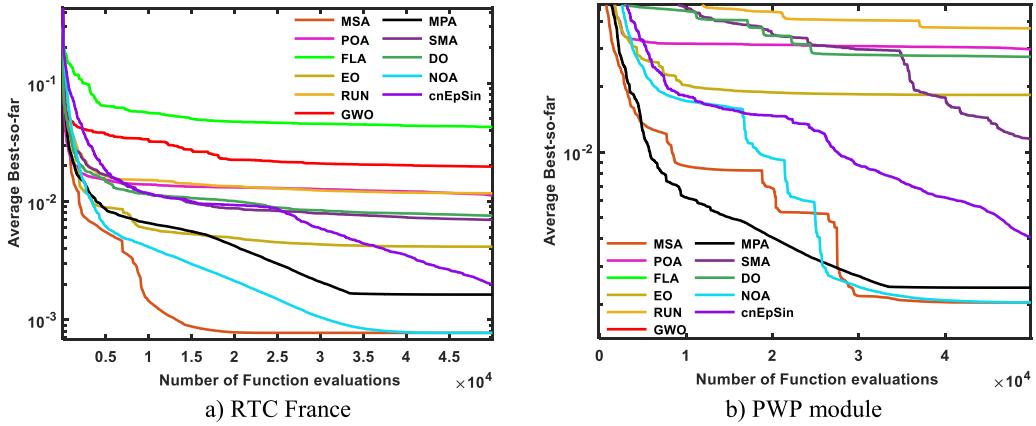
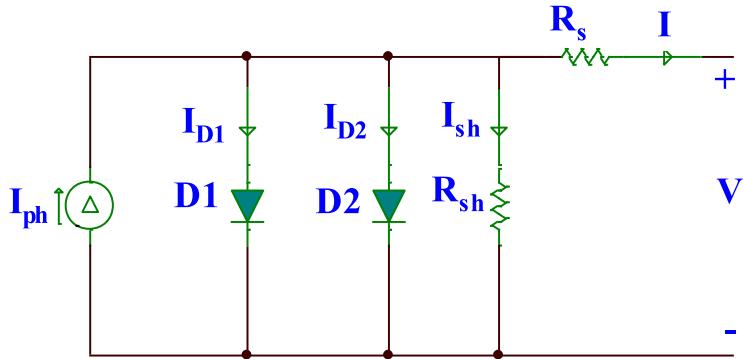
This equation could be extended as follows:

$$I = I_{ph} - I_{sd1}(\exp\left(\frac{V + I * R_s}{n_1 * V_t}\right) - 1) - I_{sd2}(\exp\left(\frac{V + I * R_s}{n_2 * V_t}\right) - 1) - \frac{V + I * R_s}{R_{sh}}$$

Table 17

Comparison under SDM.

	RTC France solar cell					Photowatt-PWP201 module				
	Best	Worst	Ave	SD	p-value	Best	Worst	Ave	SD	p-value
MSA	7.73006E-04	7.73006E-04	7.73006E-04	1.94E-17		2.03999E-03	2.03999E-03	2.03999E-03	2.26E-16	
POA	3.66196E-03	7.36945E-02	1.12159E-02	1.70E-02	3.016E-11	2.81774E-03	9.51542E-02	2.96547E-02	4.02E-02	2.99E-11
FLA	6.56958E-03	1.18650E-01	4.26568E-02	4.38E-02	3.018E-11	3.82018E-03	1.63136E-01	6.82056E-02	5.63E-02	3.01E-11
EO	7.78041E-04	7.36945E-02	4.14278E-03	1.32E-02	3.018E-11	2.04654E-03	9.51542E-02	1.83552E-02	3.49E-02	3.01E-11
RUN	7.01095E-03	1.95567E-02	1.17316E-02	4.67E-03	3.018E-11	2.11747E-03	9.51542E-02	3.71145E-02	3.89E-02	3.01E-11
GWO	5.52086E-03	7.36949E-02	1.97161E-02	2.21E-02	3.018E-11	2.86259E-03	9.51542E-02	6.57086E-02	4.24E-02	3.01E-11
MPA	8.77786E-04	5.41396E-03	1.62940E-03	8.62E-04	3.018E-11	2.03999E-03	3.26871E-03	2.38633E-03	3.62E-04	1.41E-09
SMA	3.55153E-03	8.78752E-03	6.99043E-03	1.26E-03	3.018E-11	2.48233E-03	9.51542E-02	1.14910E-02	1.64E-02	3.01E-11
DO	9.13128E-04	7.36945E-02	7.59286E-03	1.27E-02	3.018E-11	3.08910E-03	9.51542E-02	2.74722E-02	3.80E-02	3.01E-11
NOA	7.73006E-04	7.73006E-04	7.73006E-04	2.38E-11	1.694E-09	2.03999E-03	2.04864E-03	2.04042E-03	1.71E-06	3.25E-05
cnEpSin	7.73006E-04	3.38180E-03	1.99435E-03	9.39E-04	6.691E-11	2.06697E-03	5.89528E-03	4.04654E-03	1.21E-03	3.01E-11

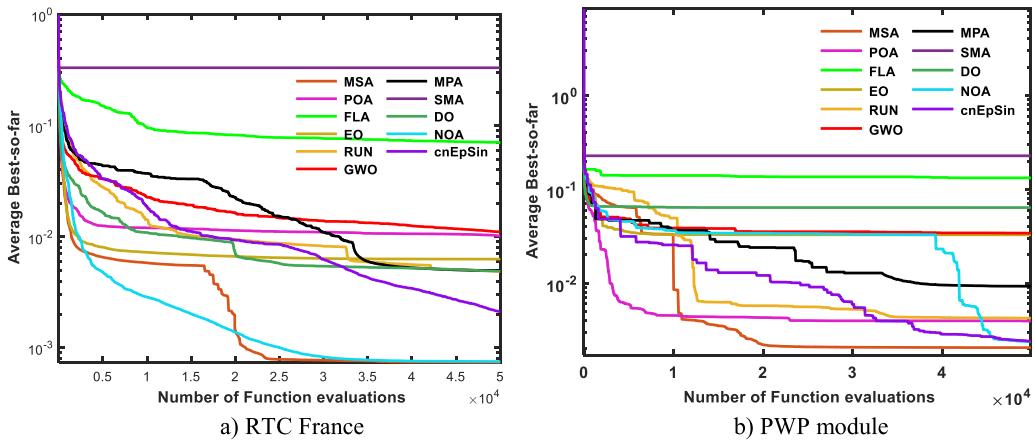
Bold values represent the best findings**Fig. 16.** Comparison in terms of convergence speed on SDM.**Fig. 17.** DDM's electrical circuit.

where I_{sd1} represents the first diode's current and I_{sd2} is the second diode's current. n_1 and n_2 are two factors to represent the ideality of the first and second diode. This model includes five unknown parameters (I_{ph} , I_{sd1} , I_{sd2} , R_s , R_{sh} , n_1 , and n_2) that have to be accurately identified to emulate the DDM accurately. To estimate them, MSA and the compared algorithms are executed 30 independent times for both the RTC France cell and PWP module, and the various performance metrics are computed and presented in Table 18. This table shows the superiority of MSA, where it could be the best for all the performance measures. To show the difference between the outcomes of

Table 18

Comparison under DDM.

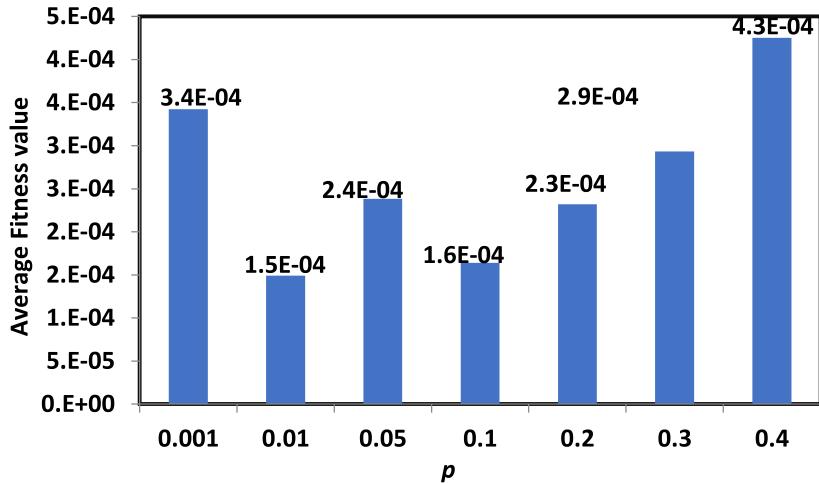
RTC France solar cell					Photowatt-PWP201 module						
	Best	Worst	Ave	SD	p-value		Best	Worst	Ave	SD	p-value
MSA	7.32648E-04	7.69885E-04	7.42703E-04	6.24E-06		2.04040E-03	2.06233E-03	2.05356E-03	1.16E-05		
POA	1.29361E-03	6.94425E-02	1.49424E-02	2.48E-02	3.02E-11	3.26309E-03	4.42975E-03	3.94194E-03	6.06E-04	3.02E-11	
FLA	2.58298E-03	1.29412E-01	5.83534E-02	4.66E-02	3.02E-11	1.05253E-01	1.50039E-01	1.32277E-01	2.38E-02	3.02E-11	
EO	7.43375E-04	6.94424E-02	1.06494E-02	2.35E-02	1.33E-10	2.13918E-03	9.39027E-02	3.27301E-02	5.30E-02	8.99E-11	
RUN	1.05159E-03	7.65295E-02	1.07835E-02	1.81E-02	3.02E-11	3.87488E-03	4.57543E-03	4.21940E-03	3.50E-04	3.02E-11	
GWO	8.38582E-04	6.96273E-02	1.32661E-02	2.29E-02	3.02E-11	3.38300E-03	9.39074E-02	3.40536E-02	5.18E-02	3.02E-11	
MPA	1.32590E-03	1.14799E-02	5.58696E-03	2.38E-03	3.02E-11	6.79367E-03	1.10611E-02	9.26556E-03	2.21E-03	3.02E-11	
SMA	1.35261E-01	4.38873E-01	3.19707E-01	6.64E-02	3.02E-11	2.13272E-01	2.43053E-01	2.27309E-01	1.50E-02	3.02E-11	
DO	7.70812E-04	6.94425E-02	1.37900E-02	2.54E-02	3.34E-11	3.61922E-03	9.39028E-02	6.38083E-02	5.21E-02	3.02E-11	
NOA	7.38218E-04	7.75975E-04	7.57343E-04	7.89E-06	6.53E-08	2.06638E-03	2.94463E-03	2.37024E-03	4.98E-04	2.07E-02	
cnEpSin	7.32648E-04	4.14322E-03	1.69916E-03	1.33E-03	4.12E-05	2.21965E-03	2.61532E-03	2.38862E-03	2.04E-04	1.12E-02	

Bold values represent the best findings.**Fig. 18.** Comparison in terms of convergence speed on DDM.

MSA and those of the other algorithms, the p -value between MSA and each compared algorithm is computed and presented in the same table, which shows that MSA produces significantly-different outcomes. In addition, Fig. 18 depicts the convergence curves of MSA and the other algorithms on both RTC France and the PWP module. From this figure, MSA converges faster than the others.

7. Application III: Parameters selection of PEMFCs

In this section, an additional application is used to further evaluate the performance of MSA in comparison to several rival optimizers under considered performance metrics. This application is the parameter estimation of the proton exchange membrane fuel cell (PEMFCs). The mathematical model of this cell, described in detail in [136], involves seven unknown parameters ($\xi_1, \xi_2, \xi_3, \xi_4, \lambda, R_c$, and β) that need to be accurately estimated to maximize its performance. The compared algorithms involve interior search algorithm (ISA) [137], artificial ecosystem-based optimization (AEO) [136], moth-flame optimizer (MFO) [136], SMA [138], differential evolution (DE) [139], salp swarm algorithm (SSA) [136], WOA [69], cnEpSin, and MPA [140]. All those algorithms are validated using four well-known commercial PEMFC stacks, including 500 W stack [141], Ballard Mark V5 [141], 250 W stack [141], and Avista SR-12 [141], under the same parameters recommended in the published papers, with the exception of the population size and the maximum number of function evaluations, which are set to 15 and 3000 to ensure a fair comparison. The characteristics of those PEMFC stacks are extensively described in [142,143]. The sum of the squared error (SSE) between the measured and estimated voltages are considered as the objective function that needs to be solved by MSA to find the unknown parameters of PEMFC. The mathematical model of this function

**Fig. 19.** Tuning the parameter p .

is formulated as follows:

$$SSE = f(\vec{x}_i^t) = \sum_{c=1}^M |V_{S,measured}(c) - V_{S,estimated}(c)|^2$$

The proposed MSA has six controlling parameters that need to be accurately tuned to maximize its performance when solving this application. After conducting several experiments, we found that its performance is strong when setting those parameters to the same values estimated in Section 4.1, except for the parameter p , which could improve MSA's performance when it is set to 0.01 as described in Fig. 19. Ultimately, the lower and upper bounds of each unknown parameter used in our experiments are described below according to several papers in the literature [136,141]:

$$\begin{aligned} -1.1997 &< \xi_1 < -0.8532 \\ 0.001 &< \xi_2 < 0.005 \\ 3.6e-5 &< \xi_3 < 9.8e-5 \\ -2.6e-4 &< \xi_4 < -9.54e-5 \\ 13.000 &< \lambda < 23.000 \\ 1e-4 &< R_c < 8e-4 \\ 1.36e-2 &< \beta < 50e-2 \end{aligned}$$

The proposed MSA and rival optimizers executed 25 independent runs on each considered PEMFC stack, the obtained SSE values obtained within those runs are analyzed in terms of Best, Worst, Ave, SD, and p -value, as defined in Table 19. Inspecting this table reveals that MSA could achieve the best value for all used metrics on all considered PEMFC stacks. Also, the p -values presented in this table reveal that the outcomes produced by MSA are significantly different from those of the rival optimizers. To further reveal the MSA's superiority, the convergence curve on each PEMFC stack is estimated and presented in Fig. 20; this figure shows that MSA converges faster than all the compared algorithms.

8. Conclusions and future research perspectives

A novel nature-inspired metaheuristic optimization algorithm inspired by the hunting behavior of praying mantises was proposed. The proposed optimizer, namely, the mantis search optimization algorithm (MSA), involves three optimization operators: search for prey (exploration), attack prey (exploitation), and sexual cannibalism to further improve the exploration and exploitation operators. Two mathematical benchmarks were adopted. The first

Table 19

Comparison for various considered PEMFC stacks.

BCS 500 W stack					250 W stack						
	Best	Worst	Ave	SD	p-value		Best	Worst	Ave	SD	p-value
MSA	1.16984E-02	1.28536E-02	2.09209E-02	2.24333E-03		3.3598E-01	3.3662E-01	3.3983E-01	8.3498E-04		
NOA	1.51860E-02	1.00777E-01	2.95559E-01	8.72919E-02	7.3E-11	3.3699E-01	4.0547E-01	1.1845E+00	1.6144E-01	8.9E-11	
DE	5.77747E+00	5.93127E+00	7.12563E+00	2.61322E-01	3.0E-11	1.9154E+00	1.9742E+00	2.0482E+00	4.7128E-02	3.0E-11	
SMA	1.18890E-02	7.42946E-02	2.98176E-01	6.96626E-02	4.2E-10	3.3621E-01	4.0431E-01	8.5613E-01	1.0733E-01	5.0E-10	
MPA	4.72949E-02	3.34690E+00	1.43592E+01	3.15504E+00	3.0E-11	3.5097E-01	1.5452E+00	2.6195E+00	6.6731E-01	3.0E-11	
MFO	5.77559E+00	6.44108E+00	1.08734E+01	1.18063E+00	2.9E-11	1.9144E+00	2.0765E+00	3.7097E+00	4.4762E-01	3.0E-11	
SSA	5.22070E+00	6.73368E+00	9.28624E+00	1.10072E+00	3.0E-11	1.7159E+00	2.0447E+00	3.8159E+00	4.3133E-01	3.0E-11	
WOA	3.81986E-01	4.71318E+00	2.97576E+01	5.74898E+00	3.0E-11	4.5910E-01	2.8519E+00	9.2251E+00	2.0837E+00	3.0E-11	
ISA	1.18437E-02	3.92459E-01	5.59629E+00	1.03996E+00	4.2E-10	3.3622E-01	4.2194E-01	1.6047E+00	2.7468E-01	7.0E-08	
cnEpSin	1.17356E-02	2.14760E-02	1.12558E-01	1.90538E-02	2.7E-05	3.3620E-01	3.3934E-01	3.5705E-01	3.9518E-03	5.6E-07	
Ballard Mark V stack					SR-12 stack						
	Best	Worst	Ave	SD	p-value		Best	Worst	Ave	SD	p-value
MSA	8.5363E-01	8.9321E-01	1.1317E+00	7.1843E-02		1.4210E-04	6.4172E-04	1.2913E-02	2.3235E-03		
NOA	9.2428E-01	1.7099E+00	1.0484E+01	1.7017E+00	5.1E-10	2.0355E-03	1.0081E-01	7.1679E-01	1.4438E-01	4.1E-11	
DE	1.7660E+00	1.8100E+00	2.0931E+00	7.2753E-02	3.0E-11	1.1213E-01	4.9878E-01	3.7621E+00	6.8193E-01	3.0E-11	
SMA	8.5824E-01	1.1252E+00	1.6076E+00	2.2064E-01	1.7E-07	6.9252E-04	4.1167E-02	2.1818E-01	4.7695E-02	8.2E-11	
MPA	9.6079E-01	2.1140E+00	3.6427E+00	6.7271E-01	5.0E-11	5.7472E-02	4.0440E+00	2.5850E+01	5.4696E+00	3.0E-11	
MFO	1.7656E+00	3.5557E+00	5.3910E+01	9.5115E+00	3.0E-11	1.0271E-01	1.4550E-01	4.4845E-01	8.1368E-02	3.0E-11	
SSA	1.7464E+00	2.0934E+00	3.1066E+00	3.2115E-01	3.0E-11	1.0068E-01	1.8831E-01	3.5225E-01	8.4639E-02	3.0E-11	
WOA	1.4128E+00	5.9208E+00	6.3125E+01	1.1642E+01	3.0E-11	1.3517E-02	1.6589E+01	1.2854E+02	3.8102E+01	3.0E-11	
ISA	8.5361E-01	1.2280E+00	2.2642E+00	4.0880E-01	7.7E-05	2.8122E-04	2.4984E-02	2.9617E-01	5.5024E-02	6.1E-10	
cnEpSin	8.5415E-01	9.0196E-01	1.2446E+00	8.5479E-02	4.1E-02	2.3504E-04	1.3799E-02	1.6381E-01	3.0138E-02	2.7E-09	

Bold values represent the best outcomes.

one involved 23 unimodal and multimodal test functions (6 unimodal and 17 multimodal), and the second one was CEC-2017 which included four families of mathematical test functions (unimodal, multimodal, composition, and hybrid) with a total of 29 test functions. These benchmarks were used to validate the efficiency of the introduced optimizer. The results obtained by MSA on the two benchmarks were extensively compared with those of nine well-known, well-established optimizers to show MSA's effectiveness. MSA was superior to its rival optimizers.

To further articulate the efficiency of the proposed MSA, three real-world applications: five engineering design problems and the parameter estimation problem of photovoltaic modules and fuel cells were solved. MSA was superior to the compared algorithms in several problems and showed competitiveness with several recently published ones. Our future research perspectives involve proposing multi-objective and binary versions of the MSA algorithm, which are currently under development.

Funding

The authors have no funding for this research.

Human and animal rights

This article does not contain any studies with human or animal subjects performed by any of the authors.

Informed consent

It was obtained from all individual participants included in the study.

CRediT authorship contribution statement

Mohamed Abdel-Basset: Investigation, Conceptualization, Methodology, Visualization, Resources, Validation, Software, Writing – review & editing, Writing – original draft. **Reda Mohamed:** Investigation, Conceptualization, Methodology, Visualization, Resources, Validation, Software, Writing – review & editing, Writing – original draft. **Mahinda Zidan:** Conceptualization, Methodology, Visualization, Resources, Validation, Writing – review & editing.

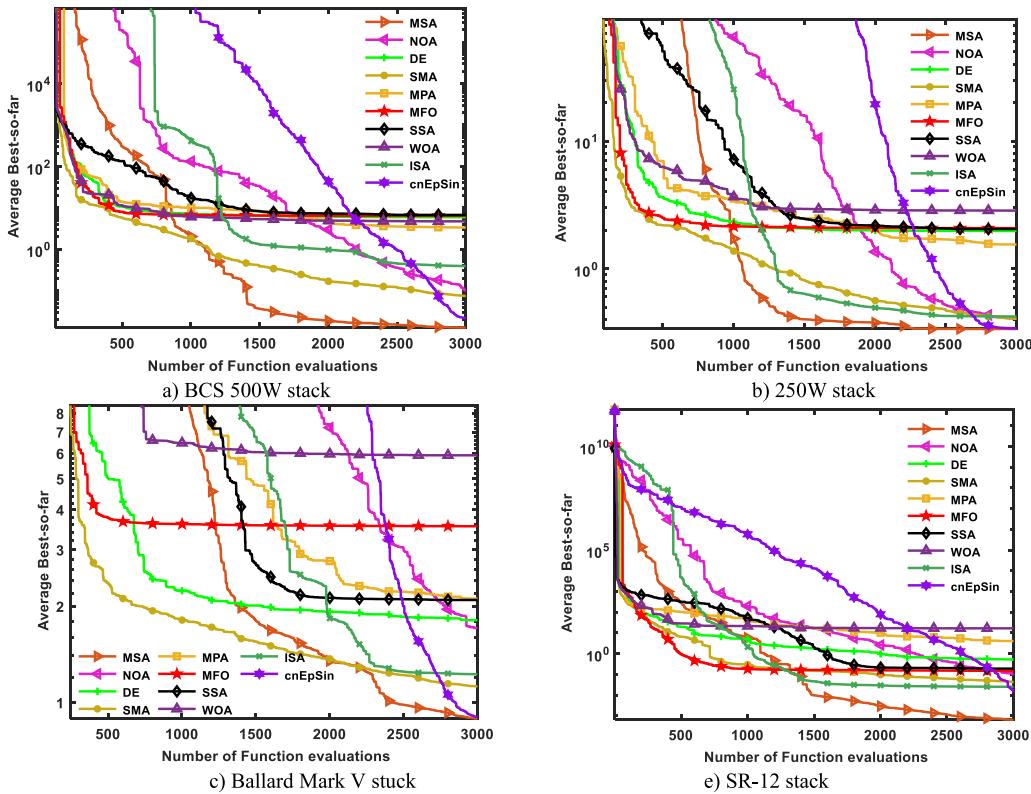


Fig. 20. Comparison among algorithms using convergence curve for each PEMFC stack.

Mohammed Jameel: Conceptualization, Methodology, Visualization, Resources, Validation, Writing – review & editing. **Mohamed Abouhawwash:** Conceptualization, Methodology, Visualization, Resources, Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

- [1] W. Zhao, L. Wang, S. Mirjalili, Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications, *Comput. Methods Appl. Mech. Engrg.* 388 (2022) 114194.
- [2] Z. Beheshti, S.M.H. Shamsuddin, A review of population-based meta-heuristic algorithms, *Int. J. Adv. Soft Comput. Appl.* 5 (1) (2013) 1–35.
- [3] B. Abdollahzadeh, F. Soleimanian Gharehchopogh, S. Mirjalili, Artificial gorilla troops optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems, *Int. J. Intell. Syst.* 36 (10) (2021) 5887–5958.
- [4] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, *IEEE Trans. Evol. Comput.* 3 (2) (1999) 82–102.
- [5] T. Mantere, J.T. Alander, Evolutionary software engineering, a review, *Appl. Soft Comput.* 5 (3) (2005) 315–331.
- [6] S. Yang, et al., A general multi-objective optimized wavelet filter and its applications in fault diagnosis of wheelset bearings, *Mech. Syst. Signal Process.* 145 (2020) 106914.
- [7] W. Zhao, L. Wang, Z. Zhang, Atom search optimization and its application to solve a hydrogeologic parameter estimation problem, *Knowl.-Based Syst.* 163 (2019) 283–304.

- [8] H. Wang, et al., Solving team making problem for crowdsourcing with hybrid metaheuristic algorithm, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, 2018.
- [9] W. Zhao, L. Wang, Z. Zhang, Supply–demand-based optimization: a novel economics-inspired algorithm for global optimization, *IEEE Access* 7 (2019) 73182–73206.
- [10] E. Rodríguez-Esparza, et al., An efficient Harris hawks-inspired image segmentation method, *Expert Syst. Appl.* 155 (2020) 113428.
- [11] I. Abu Doush, E. Santos, A sensitivity analysis for harmony search with multi-parent crossover algorithm, in: Proceedings of SAI Intelligent Systems Conference, Springer, 2019.
- [12] N. Ferro, S. Micheletti, S. Perotto, An optimization algorithm for automatic structural design, *Comput. Methods Appl. Mech. Engrg.* 372 (2020) 113335.
- [13] L. Abualigah, et al., The arithmetic optimization algorithm, *Comput. Methods Appl. Mech. Engrg.* 376 (2021) 113609.
- [14] W. Zhao, et al., An adaptive hybrid atom search optimization with particle swarm optimization and its application to optimal no-load PID design of hydro-turbine governor, *J. Comput. Des. Eng.* 8 (5) (2021) 1204–1233.
- [15] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Softw.* 69 (2014) 46–61.
- [16] S. Shadravan, H.R. Naji, V.K. Bardsiri, The sailfish optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems, *Eng. Appl. Artif. Intell.* 80 (2019) 20–34.
- [17] K.M. Ong, P. Ong, C.K. Sia, A carnivorous plant algorithm for solving global optimization problems, *Appl. Soft Comput.* 98 (2021) 106833.
- [18] S. Saremi, S. Mirjalili, A. Lewis, Grasshopper optimisation algorithm: theory and application, *Adv. Eng. Softw.* 105 (2017) 30–47.
- [19] A. Cheraghaliour, M. Hajaghaei-Keshteli, M.M. Paydar, Tree growth algorithm (TGA): A novel approach for solving optimization problems, *Eng. Appl. Artif. Intell.* 72 (2018) 393–414.
- [20] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, *IEEE Trans. Evol. Comput.* 1 (1) (1997) 67–82.
- [21] J.H. Holland, Genetic algorithms, *Sci. Am.* 267 (1) (1992) 66–73.
- [22] J.R. Koza, Genetic programming as a means for programming computers by natural selection, *Stat. Comput.* 4 (2) (1994) 87–112.
- [23] K.V. Price, Differential evolution, in: *Handbook of Optimization*, Springer, 2013, pp. 187–214.
- [24] D. Simon, Biogeography-based optimization, *IEEE Trans. Evol. Comput.* 12 (6) (2008) 702–713.
- [25] H.-G. Beyer, H.-P. Schwefel, Evolution strategies—a comprehensive introduction, *Nat. Comput.* 1 (1) (2002) 3–52.
- [26] Y. Cao, Q. Wu, Evolutionary programming, in: Proceedings of 1997 IEEE International Conference on Evolutionary Computation, ICEC'97, IEEE, 1997.
- [27] R.-J. Kuo, F.E. Zulvia, The gradient evolution algorithm: A new metaheuristic, *Inform. Sci.* 316 (2015) 246–265.
- [28] M. Ghaemi, M.-R. Feizi-Derakhshi, Forest optimization algorithm, *Expert Syst. Appl.* 41 (15) (2014) 6676–6687.
- [29] M.S. Kiran, TSA: Tree-seed algorithm for continuous optimization, *Expert Syst. Appl.* 42 (19) (2015) 6686–6698.
- [30] S. Kirkpatrick, C.D. Gelatt Jr., M.P. Vecchi, Optimization by simulated annealing, *Science* 220 (4598) (1983) 671–680.
- [31] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: A gravitational search algorithm, *Inform. Sci.* 179 (13) (2009) 2232–2248.
- [32] A. Kaveh, S. Talatahari, A novel heuristic optimization method: charged system search, *Acta Mech.* 213 (3) (2010) 267–289.
- [33] O.K. Erol, I. Eksin, A new optimization method: big bang–big crunch, *Adv. Eng. Softw.* 37 (2) (2006) 106–111.
- [34] M. Abdel-Basset, et al., Kepler optimization algorithm: A new metaheuristic algorithm inspired by Kepler's laws of planetary motion, *Knowl.-Based Syst.* (2023) 110454.
- [35] A. Hatamlou, Black hole: A new heuristic optimization approach for data clustering, *Inform. Sci.* 222 (2013) 175–184.
- [36] H. Su, et al., RIME: A physics-based optimization, *Neurocomputing* 532 (2023) 183–214.
- [37] H. Shah-Hosseini, Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation, *Int. J. Comput. Sci. Eng.* 6 (1–2) (2011) 132–140.
- [38] F.A. Hashim, et al., Henry gas solubility optimization: A novel physics-based algorithm, *Future Gener. Comput. Syst.* 101 (2019) 646–667.
- [39] A. Kaveh, M. Khayatazad, A new meta-heuristic method: ray optimization, *Comput. Struct.* 112 (2012) 283–294.
- [40] R.A. Formato, Central force optimization, *Prog. Electromagn. Res.* 77 (1) (2007) 425–491.
- [41] B. Alatas, ACROA: artificial chemical reaction optimization algorithm for global optimization, *Expert Syst. Appl.* 38 (10) (2011) 13170–13180.
- [42] S. Mirjalili, SCA: a sine cosine algorithm for solving optimization problems, *Knowl.-Based Syst.* 96 (2016) 120–133.
- [43] H. Eskandar, et al., Water cycle algorithm—A novel metaheuristic optimization method for solving constrained engineering optimization problems, *Comput. Struct.* 110 (2012) 151–166.
- [44] H. Du, X. Wu, J. Zhuang, Small-world optimization algorithm for function optimization, in: *International Conference on Natural Computation*, Springer, 2006.
- [45] S. Mirjalili, S.M. Mirjalili, A. Hatamlou, Multi-verse optimizer: a nature-inspired algorithm for global optimization, *Neural Comput. Appl.* 27 (2) (2016) 495–513.
- [46] S.I. Birbil, S.-C. Fang, An electromagnetism-like mechanism for global optimization, *J. Global Optim.* 25 (3) (2003) 263–282.
- [47] H. Shah-Hosseini, The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm, *Int. J. Bio-Inspired Comput.* 1 (1–2) (2009) 71–79.
- [48] B. Javidy, A. Hatamlou, S. Mirjalili, Ions motion algorithm for solving optimization problems, *Appl. Soft Comput.* 32 (2015) 72–79.
- [49] C.-L. Chuang, J.-A. Jiang, Integrated radiation optimization: inspired by the gravitational radiation in the curvature of space–time, in: *2007 IEEE Congress on Evolutionary Computation*, IEEE, 2007.
- [50] P. Rabanal, I. Rodriguez, F. Rubio, Using river formation dynamics to design heuristic algorithms, in: *International Conference on Unconventional Computation*, Springer, 2007.

- [51] Y.-T. Hsiao, et al., A novel optimization algorithm: space gravitational optimization, in: 2005 IEEE International Conference on Systems, Man and Cybernetics, IEEE, 2005.
- [52] A. Faramarzi, et al., Equilibrium optimizer: A novel optimization algorithm, *Knowl.-Based Syst.* 191 (2020) 105190.
- [53] L. Xie, J. Zeng, Z. Cui, General framework of artificial physics optimization algorithm, in: 2009 World Congress on Nature & Biologically Inspired Computing, NaBIC, IEEE, 2009.
- [54] B. Webster, P.J. Bernhard, A local search optimization algorithm based on natural principles of gravitation, 2003.
- [55] F.A. Hashim, et al., Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems, *Appl. Intell.* 51 (3) (2021) 1531–1551.
- [56] M. Abdel-Basset, et al., Young's double-slit experiment optimizer : A novel metaheuristic optimization algorithm for global and constraint optimization problems, *Comput. Methods Appl. Mech. Engrg.* 403 (2023) 115652.
- [57] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-International Conference on Neural Networks, IEEE, 1995.
- [58] S. Li, et al., Slime mould algorithm: A new method for stochastic optimization, *Future Gener. Comput. Syst.* 111 (2020) 300–323.
- [59] M.S. Braik, Chameleon swarm algorithm: A bio-inspired optimizer for solving engineering design problems, *Expert Syst. Appl.* 174 (2021) 114685.
- [60] F. Miar Naeimi, G. Azizyan, M. Rashki, Horse herd optimization algorithm: A nature-inspired algorithm for high-dimensional optimization problems, *Knowl.-Based Syst.* 213 (2021) 106711.
- [61] A. Faramarzi, et al., Marine predators algorithm: A nature-inspired metaheuristic, *Expert Syst. Appl.* 152 (2020) 113377.
- [62] M. Abdel-Basset, et al., Nutcracker optimizer: A novel nature-inspired metaheuristic algorithm for global optimization and engineering design problems, *Knowl.-Based Syst.* 262 (2023) 110248.
- [63] M. Zhang, G. Wen, J. Yang, Duck swarm algorithm: a novel swarm intelligence algorithm, 2021, arXiv preprint [arXiv:2112.13508](https://arxiv.org/abs/2112.13508).
- [64] M. Dorigo, M. Birattari, T. Stützle, Ant colony optimization, *IEEE Comput. Intell. Mag.* 1 (4) (2006) 28–39.
- [65] M. Abdel-Basset, et al., Spider wasp optimizer: a novel meta-heuristic optimization algorithm, *Artif. Intell. Rev.* (2023).
- [66] X.-S. Yang, Firefly algorithm, stochastic test functions and design optimisation, *Int. J. Bio-Inspired Comput.* 2 (2) (2010) 78–84.
- [67] X. Lu, Y. Zhou, A novel global convergence algorithm: bee collecting pollen algorithm, in: International Conference on Intelligent Computing, Springer, 2008.
- [68] D. Polap, M. Woźniak, Red fox optimization algorithm, *Expert Syst. Appl.* 166 (2021) 114107.
- [69] S. Mirjalili, A. Lewis, The whale optimization algorithm, *Adv. Eng. Softw.* 95 (2016) 51–67.
- [70] X.S. Yang, A.H. Gandomi, Bat algorithm: a novel approach for global engineering optimization, *Eng. Comput.* (2012).
- [71] R. Oftadeh, M. Mahjoob, M. Shariatpanahi, A novel meta-heuristic optimization algorithm inspired by group hunting of animals: Hunting search, *Comput. Math. Appl.* 60 (7) (2010) 2087–2098.
- [72] S. Mirjalili, et al., Salp swarm algorithm: A bio-inspired optimizer for engineering design problems, *Adv. Eng. Softw.* 114 (2017) 163–191.
- [73] S.-C. Chu, P.-W. Tsai, J.-S. Pan, Cat swarm optimization, in: Pacific Rim International Conference on Artificial Intelligence, Springer, 2006.
- [74] A.L.a. Bolaji, et al., A comprehensive review: Krill Herd algorithm (KH) and its applications, *Appl. Soft Comput.* 49 (2016) 437–446.
- [75] A.H. Gandomi, X.-S. Yang, A.H. Alavi, Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, *Eng. Comput.* 29 (1) (2013) 17–35.
- [76] A.S. Shamsaldin, et al., Donkey and smuggler optimization algorithm: A collaborative working approach to path finding, *J. Comput. Des. Eng.* 6 (4) (2019) 562–583.
- [77] M. Dehghani, Š. Hubálovský, P. Trojovský, Northern Goshawk optimization: A new swarm-based algorithm for solving optimization problems, *IEEE Access* 9 (2021) 162059–162080.
- [78] B. Abdollahzadeh, et al., Mountain Gazelle optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems, *Adv. Eng. Softw.* 174 (2022) 103282.
- [79] G.-G. Wang, S. Deb, L.d.S. Coelho, Elephant herding optimization, in: 2015 3rd International Symposium on Computational and Business Intelligence, ISCB1, IEEE, 2015.
- [80] H. Zamani, M.H. Nadimi-Shahroki, A.H. Gandomi, Starling murmuration optimizer: A novel bio-inspired algorithm for global and engineering optimization, *Comput. Methods Appl. Mech. Engrg.* 392 (2022) 114616.
- [81] S. Arora, S. Singh, Butterfly optimization algorithm: a novel approach for global optimization, *Soft Comput.* 23 (3) (2019) 715–734.
- [82] D. Wang, D. Tan, L. Liu, Particle swarm optimization algorithm: an overview, *Soft Comput.* 22 (2) (2018) 387–408.
- [83] R.V. Rao, V.J. Savsani, D. Vakharia, Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems, *Comput. Aided Des.* 43 (3) (2011) 303–315.
- [84] N. Ghorbani, E. Babaei, Exchange market algorithm, *Appl. Soft Comput.* 19 (2014) 177–187.
- [85] Z.W. Geem, J.H. Kim, G.V. Loganathan, A new heuristic optimization algorithm: harmony search, *Simulation* 76 (2) (2001) 60–68.
- [86] Q. Askari, I. Younas, M. Saeed, Political optimizer: A novel socio-inspired meta-heuristic for global optimization, *Knowl.-Based Syst.* 195 (2020) 105709.
- [87] A. Naik, S.C. Satapathy, Past present future: a new human-based algorithm for stochastic optimization, *Soft Comput.* 25 (20) (2021) 12915–12976.
- [88] Y. Shi, Brain storm optimization algorithm, in: International Conference in Swarm Intelligence, Springer, 2011.
- [89] N. Moosavian, B.K. Roodsari, Soccer league competition algorithm: A novel meta-heuristic algorithm for optimal design of water distribution networks, *Swarm Evol. Comput.* 17 (2014) 14–24.
- [90] S. Talatahari, M. Azizi, Chaos game optimization: a novel metaheuristic algorithm, *Artif. Intell. Rev.* 54 (2) (2021) 917–1004.

- [91] F.R. Prete, et al., Visual stimuli that elicit visual tracking, approaching and striking behavior from an unusual praying mantis, *Euchomenella macrops* (Insecta: Mantodea), *J. Insect Physiol.* 58 (5) (2012) 648–659.
- [92] H. Maldonado, A learning process in the praying mantis, *Physiol. Behav.* 9 (3) (1972) 435–445.
- [93] A. Nichols, The Praying Mantis (The Encounter), in: SWOSU Sayre Student Anthology, vol. 1, no. 5, 2019, p. 31.
- [94] V. Nityananda, et al., Motion-in-depth perception and prey capture in the praying mantis, 2018, 502583, bioRxiv.
- [95] A.P. Getsy, The Development of a Visual System for MantisBot: A Robot Modeled After the Praying Mantis, Case Western Reserve University, 2016.
- [96] R.J. Pearce, Praying mantis: A unique Glen Meyer village in London, *Ont. Archaeol.* 85 (88) (2008) 97–120.
- [97] S. Brannoch, et al., Manual of praying mantis morphology, nomenclature, and practices (Insecta, Mantodea), *ZooKeys* 696 (2017) 1–100.
- [98] C.E. Oufiero, Ontogenetic changes in behavioral and kinematic components of prey capture strikes in a praying mantis, *Evol. Ecol.* (2021) 1–19.
- [99] C.E. Oufiero, Evolutionary diversification in the raptorial forelegs of Mantodea: relations to body size and depth perception, *J. Morphol.* 281 (4–5) (2020) 513–522.
- [100] G. Waldbauer, How Not to Be Eaten: The Insects Fight Back, University of California Press, 2021.
- [101] G.J. Svenson, C. Medellin, C.E. Sarmiento, Re-evolution of a morphological precursor of crypsis investment in the newly revised horned praying mantises (Insecta, Mantodea, Vatinae), *Sys. Entomol.* 41 (1) (2016) 229–255.
- [102] T. Birkhead, K. Lee, P. Young, Sexual cannibalism in the praying mantis *Hierodula membranacea*, *Behaviour* (1988) 112–118.
- [103] J. Rivera, Y. Callohuari, A new species of praying mantis from Peru reveals impaling as a novel hunting strategy in Mantodea (Thespidae: Thespini), *Neotropical Entomol.* 49 (2) (2020) 234–249.
- [104] P. Prokop, R. Václav, Seasonal aspects of sexual cannibalism in the praying mantis (*Mantis religiosa*), *J. Ethol.* 26 (2) (2008) 213–218.
- [105] J.P. Lelito, W.D. Brown, Complicity or conflict over sexual cannibalism? Male risk taking in the praying mantis *Tenodera aridifolia sinensis*, *Amer. Nat.* 168 (2) (2006) 263–269.
- [106] T. Iwasaki, Predatory behavior of the praying mantis, *Tenodera aridifolia* I. Effect of prey size on prey recognition, *J. Ethol.* 8 (2) (1990) 75–79.
- [107] H. Maldonado, L. Levin, J. Pita, Hit distance and the predatory strike of the praying mantis, *Z. Vergleichende Physiol.* 56 (3) (1967) 237–257.
- [108] B.J. Corrette, Prey capture in the praying mantis *Tenodera aridifolia sinensis*: coordination of the capture sequence and strike movements, *J. Exp. Biol.* 148 (1) (1990) 147–180.
- [109] G.J. Svenson, M.F. Whiting, Reconstructing the origins of praying mantises (Dictyoptera, Mantodea): the roles of Gondwanan vicariance and morphological convergence, *Cladistics* 25 (5) (2009) 468–514.
- [110] V. Nityananda, et al., Insect stereopsis demonstrated using a 3D insect cinema, *Sci. Rep.* 6 (1) (2016) 1–9.
- [111] K. Kral, The functional significance of mantis peering behaviour, *Eur. J. Entomol.* 109 (3) (2012).
- [112] C. Ruiz, J. Theobald, Insect vision: Judging distance with binocular motion disparities, *Curr. Biol.* 28 (4) (2018) R148–R150.
- [113] C.F. Michaels, S. Prindle, M. Turvey, A note on the natural basis of action categories: The catching distance of mantids, *J. Motor Behav.* 17 (2) (1985) 255–264.
- [114] K. Kral, M. Poteser, Relationship between body size and spatial vision in the praying mantis-An ontogenetic study, *J. Orthoptera Res.* (2009) 153–158.
- [115] Y. Yamawaki, et al., Coordinated movements of the head and body during orienting behaviour in the praying mantis *Tenodera aridifolia*, *J. Insect Physiol.* 57 (7) (2011) 1010–1016.
- [116] J.P. Lelito, W.D. Brown, Mate attraction by females in a sexually cannibalistic praying mantis, *Behav. Ecol. Sociobiol.* 63 (2) (2008) 313–320.
- [117] B. Abdollahzadeh, F.S. Gharehchopogh, S. Mirjalili, African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems, *Comput. Ind. Eng.* 158 (2021) 107408.
- [118] I. Ahmadianfar, O. Bozorg-Haddad, X. Chu, Gradient-based optimizer: A new metaheuristic optimization algorithm, *Inform. Sci.* 540 (2020) 131–159.
- [119] I. Ahmadianfar, et al., INFO: An efficient optimization algorithm based on weighted mean of vectors, *Expert Syst. Appl.* (2022) 116516.
- [120] I. Ahmadianfar, et al., RUN beyond the metaphor: an efficient optimization algorithm based on Runge Kutta method, *Expert Syst. Appl.* 181 (2021) 115079.
- [121] G. Wu, R. Mallipeddi, P.N. Suganthan, Problem Definitions and Evaluation Criteria for the CEC 2017 Competition on Constrained Real-Parameter Optimization, Technical Report, National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, 2017.
- [122] C.A.C. Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, *Comput. Methods Appl. Mech. Engrg.* 191 (11–12) (2002) 1245–1287.
- [123] J. Arora, Introduction to Optimum Design, Elsevier, 2004.
- [124] C.A.C. Coello, E.M. Montes, Constraint-handling in genetic algorithms through the use of dominance-based tournament selection, *Adv. Eng. Inform.* 16 (3) (2002) 193–203.
- [125] M.-Y. Cheng, D. Prayogo, Symbiotic organisms search: a new metaheuristic optimization algorithm, *Comput. Struct.* 139 (2014) 98–112.
- [126] T. Easwarakhanthan, et al., Nonlinear minimization algorithm for determining the solar cell parameters with microcomputers, *Int. J. Sol. Energy* 4 (1) (1986) 1–12.

- [127] M. Abdel-Basset, et al., An improved artificial jellyfish search optimizer for parameter identification of photovoltaic models, *Energies* 14 (7) (2021) 1867.
- [128] M.A. Soliman, H.M. Hasanien, A. Alkuhayli, Marine predators algorithm for parameters identification of triple-diode photovoltaic models, *IEEE Access* 8 (2020) 155832-155842.
- [129] S. Zhao, et al., Dandelion optimizer: A nature-inspired metaheuristic algorithm for engineering applications, *Eng. Appl. Artif. Intell.* 114 (2022) 105075.
- [130] F.A. Hashim, et al., Fick's law algorithm: A physical law-based algorithm for numerical optimization, *Knowl.-Based Syst.* 260 (2023) 110146.
- [131] C. Kumar, et al., A new stochastic slime mould optimization algorithm for the estimation of solar photovoltaic cell parameters, *Optik* 223 (2020) 165277.
- [132] P. Trojovský, M. Dehghani, Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications, *Sensors* 22 (3) (2022) 855.
- [133] N.H. Awad, M.Z. Ali, P.N. Suganthan, Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems, in: 2017 IEEE Congress on Evolutionary Computation, CEC, IEEE, 2017.
- [134] Y.T. Tan, D.S. Kirschen, N. Jenkins, A model of PV generation suitable for stability analysis, *IEEE Trans. Energy Convers.* 19 (4) (2004) 748–755.
- [135] A. Askarzadeh, A. Rezazadeh, Parameter identification for solar cell models using harmony search-based algorithms, *Sol. Energy* 86 (11) (2012) 3241–3249.
- [136] M. Abdel-Basset, et al., Adaptive and efficient optimization model for optimal parameters of proton exchange membrane fuel cells: A comprehensive analysis, *Energy* 233 (2021) 121096.
- [137] A.H. Gandomi, Interior search algorithm (ISA): a novel approach for global optimization, *ISA Trans.* 53 (4) (2014) 1168–1183.
- [138] J. Gupta, P. Nijhawan, S. Ganguli, Optimal parameter estimation of PEM fuel cell using slime mould algorithm, *Int. J. Energy Res.* 45 (10) (2021) 14732–14744.
- [139] R. Storn, K. Price, Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces, *J. Global Optim.* 11 (4) (1997) 341.
- [140] A.A.Z. Diab, et al., Fuel cell parameters estimation via marine predators and political optimizers, *IEEE Access* 8 (2020) 166998-167018.
- [141] M. Abdel-Basset, et al., Improved meta-metaheuristic algorithms for optimal parameters selection of proton exchange membrane fuel cells: A comparative study, *IEEE Access* (2023).
- [142] A.A. El-Fergany, Electrical characterisation of proton exchange membrane fuel cells stack using grasshopper optimiser, *IET Renew. Power Gener.* 12 (1) (2018) 9–17.
- [143] A. Fathy, et al., Proton exchange membrane fuel cell model parameters identification using chaotically based-bonobo optimizer, *Energy* (2023) 126705.