Knowledge-Based Systems 243 (2022) 108457

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

White Shark Optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems

Malik Braik [a,*], Abdelaziz Hammouri [a], Jaffar Atwan [b], Mohammed Azmi Al-Betar [c,d], Mohammed A. Awadallah [e,f]

[a] Department of Computer Science, Al-Balqa Applied University, Jordan
[b] Department of Computer Information System, Al-Balqa Applied University, Jordan
[c] Artificial Intelligence Research Center (AIRC), College of Engineering and Information Technology, Ajman University, Ajman, United Arab Emirates
[d] Department of Information Technology, Al-Huson University College, Al-Balqa Applied University, Al-Huson, Irbid, Jordan
[e] Department of Computer Science, Al-Aqsa University, P.O. Box 4051, Gaza, Palestine
[f] Artificial Intelligence Research Center (AIRC), Ajman University, Ajman, United Arab Emirates

ARTICLE INFO

Article history:
Received 19 July 2021
Received in revised form 27 January 2022
Accepted 16 February 2022
Available online 22 February 2022

Keywords:
White Shark Optimizer
Meta-heuristics
Optimization
Swarm intelligence
Evolutionary computation
Nature-inspired algorithms

ABSTRACT

This paper presents a novel meta-heuristic algorithm so-called White Shark Optimizer (WSO) to solve optimization problems over a continuous search space. The core ideas and underpinnings of WSO are inspired by the behaviors of great white sharks, including their exceptional senses of hearing and smell while navigating and foraging. These aspects of behavior are mathematically modeled to accommodate a sufficiently adequate balance between exploration and exploitation of WSO and to assist search agents to explore and exploit each potential area of the search space in order to achieve optimization. The search agents of WSO randomly update their position in connection with best-so-far solutions, to eventually arrive at the optimal outcome. The performance of WSO was comprehensively benchmarked on a set of 29 test functions from the CEC-2017 test suite for several dimensions. WSO was further applied to solve the benchmark problems of the CEC-2011 evolutionary algorithm competition to prove its reliability and applicability to real-world problems. A thorough analysis of computational and convergence results was presented to shed light on the efficacy and stability levels of WSO. The performance score of WSO in terms of several statistical methods was compared with 9 well-established meta-heuristics based on the solutions generated. Friedman's and Holm's tests of the results showed that WSO revealed reasonable solutions, in terms of global optimality, avoidance of local minima and solution quality, compared to other existing meta-heuristics.

1. Introduction

Optimization is the process of finding the best combination for a set of decision variables to solve a particular problem. This process has appeared in many different fields, disciplines and myriad applications [1]. Finding a solution to optimization problems pose a norm in most of all disciplines of science and engineering [2,3], where the demand for more robust solutions is constantly increasing. This means that we need reasonable algorithms that can meet the complex nature of such up-to-date scientific and engineering difficulties. When taking a deep look at the literature on existing optimization algorithms, one can realize that there are a variety of such approaches [1,4]. These methods extend from traditional optimization methods that use either linear or non-linear programming techniques [5], to newer, nature-inspired meta-heuristic techniques [4,6], each with their own merits and demerits. Although effective in solving widely known optimization problems [7], on the one hand, traditional methods are subject to the inherent dependence on gradient information and a full of promise initial start vector is needed within the search space [7]. On the other hand, the present-day nature-inspired meta-heuristics may be very effective in addressing specific problems and may not be able to present sufficiently good solutions to other problems. This is partly due to the popular behavior of these techniques of getting entangled in local or sub-optimal solutions [4].

Difficult real-world optimization problems have appeared in a large number of real-life applications [1]. These problems are usually inherently hard due to one of these points: It is very probably that many optimization problems are intrinsically nonlinear, non-convex, involve many decision variables, and their objective

* Corresponding author.
E-mail addresses: mbraik@bau.edu.jo (M. Braik), Aziz@bau.edu.jo (A. Hammouri), Jaffaratwan@bau.edu.jo (J. Atwan), m.albetar@ajman.ac.ae, mohbetar@bau.edu.jo (M.A. Al-Betar), ma.awadallah@alaqsa.edu.ps (M.A. Awadallah).

https://doi.org/10.1016/j.knosys.2022.108457
0950-7051/

functions are, in certain cases, complicated and shackled by a variety of constraints. Further, such problems may have many local optimums and may have varied or even sharp peaks [8]. For such problems, it is vital to get under way from a decent starting point with the trust of arriving at the global optimum solution. Thus, many methods have attempted to deal with this kind of problems. These methods could be categorized as exact methods and approximate methods. Exact methods are assured to get the optimum solution in a sensible time except if the problem is categorized as an NP problem, where significant computational time is required. On the other hand, the use of approximate methods has received much interest during the past three decades. In approximate methods, the main goal is to get a favorable solution in a timely manner. Traditional optimization methods usually result in efficient optimization with the global optimum solution when implemented to solve simple engineering problems, problems of modest complexity, or problems with linear search spaces [9]. These methods usually demand elaborated knowledge about the problem of concern and might not be able to represent themselves as the best way to solve contemporary optimization problems [10,11]. In some cases, they can only locate local optimal solutions when applied to solve complex problems related to nonlinear search spaces or problems that have a big number of constraints or decision variables, however there is no guarantee that they will locate the global optimal solution [10]. Furthermore, the time required to solve the optimization problem should fall within sensible ranges [12]. However, the designer of optimization problems is continually interested in finding the global optimum solution. Practicable solutions to engineering design problems are a bunch of distinct designs with all variables' practical values. In this light, if a problem has more than one local optimal solution, the outcomes got from mathematical programming methods may rely on the adoption of an initiation point where the obtained solution may not be the global one. To counter this problem, one might commence from a primitive design, alter that design, or integrate two methods to dealing with a design problem [13]. To cope with the above issues, researchers have directed their attention towards meta-heuristics, as these algorithms have broadly revealed very promising capabilities in dealing with highly convoluted shapes of challenging optimization problems [7,14].

1.1. Meta-heuristic algorithms

In the literature, meta-heuristics can be classed as: nature-inspired versus non-nature-inspired, population-based versus single point search, dynamic objective versus static objective function, single neighborhood versus various neighborhood and memory use opposed to memory less methods [1]. Moreover, meta-heuristics can be split based on three key trends: improvement of existing methods by ruling their control parameters, hybridization of different methods to take advantage of each of them, and launch of a new method. On this one, meta-heuristics can be categorized based on the source of inspiration. This has been adopted on the basis of true simulations and principles with stochasticity to mimic a number of characteristics of biological behavior, natural life of creatures, human behavior or physical phenomena present in nature [14]. On this basis, any nature-inspired meta-heuristic algorithm can fall into one of the following four main classes:

Evolutionary-based Algorithms (EAs): this type of meta-heuristics is the most common and the oldest that mimics the concepts of evolutionary behavior of creatures in nature by relying on the concept of survival of the fittest. In EAs, there is an initial random population that evolves over generations to build new solutions and get rid of the worst ones in order to mend the fitness value. These algorithms commonly perform well in finding optimal or near optimal solutions given that they do not make any credibility about the basic fitness landscape. The list of EAs includes, but is not limited to, those presented in Table 1.

Swarm Intelligence-based (SI) algorithms: this type of algorithms is the leading class of meta-heuristics that simulates the collective, dynamic, intelligent, and concerted gregarious conduct of collections of flocks or communities found in nature. Such communities include groups of birds, a school of fish, colonies of insects such as colonies of bees and ants, flocks of animals and many flocks of other species of creatures [6]. In SI, each individual has its own wit and behavior, but a combination of individuals in the algorithms offers more power to tackle complex optimization problems. This type of meta-heuristics has gained more interest from researchers and is competing strongly with EAs. This class includes a big variety of algorithms, with some of the widely known ones shown in Table 2.

Human-based algorithms: this category characterizes the phenomena related to human behavior, non-physical activities such as thinking and their perceptions in societies [1,14]. The algorithms in this class have begun to attract researchers' attention as a new tendency in the past decade and present, but they still cannot compete with EAs and SI algorithms. Previous works in this area include a good number of methods, but we have listed the most common ones in Table 3.

Physics-based (PB) algorithms: these algorithms are derived from the dominant physical fundamentals found in the universe as a whole [74]. These algorithms arose from the laws of physics in nature and usually distinguish the interaction of search agents rooted in the prevailing rules of physical processes. Some of the most notable examples of PB algorithms comprise those given in Table 4.

1.2. Exploration and exploitation

Regardless of the differences between the above algorithms, they all pursue the same course of action to approximate or find the global optimal. First, optimization begins with a pool of random solutions, which require be aggregating and amending randomly, readily, and abruptly. This triggers the solutions to move globally. This stage is so-called *exploration* of the search space, where solutions gravitate to several and various regions of the search space by sudden changes [56]. The main intention of this stage is to locate the most promising areas in the search space, move away from the local optimum and get away from local optima slump. After adequate exploration, the solutions begin to alter reasonably and proceed locally in the most encouraging solutions of the search space in the potential of elevating the quality of their solutions. This stage is named *exploitation*, in which the key aim of this stage is to ameliorate the performance of the best solutions got during exploration [9]. Even though the avoidance of local optima may take place in the exploitation stage, the coverage of search region is not as wide as the exploration stage. In this case, solutions evade local solutions that are close to the global optimal. Therefore, we can conclude that the exploration and exploitation stages follow contradictory targets. The question here is when is the best time is to proceed from exploration to exploitation. It is not easy to answer this question on account of the stochastic nature of meta-heuristics and the unknown shape of the search space. Therefore, most of the meta-heuristics smoothly and continuously require search agents to pass from exploration to exploitation using accommodative strategies. A compelling recommendation for credible performance is to strike a sufficient fine-tuning balance between these two behaviors [96].

In contrast to traditional optimization methods, meta-heuristics have become surprisingly popular and have gained worthy of

Table 1
A brief review of evolutionary-based algorithms.

Algorithm	Inspiration	Year
Evolutionary Programming (EP) [15]	Finite state machine	1966
Evolution Strategy (ES) [16]	Biological evolution	1973
Genetic Algorithms (GAs) [17]	Evolutionary concepts	1975
Memetic Algorithm (MA) [18]	Darwinian principles and Dawkins notion of a meme	1989
Co-Evolutionary Algorithm (CEA) [19]	Sorting networks and data sets in a predator–prey type relationship	1990
Genetic Programming (GP) [20]	Biological evolution	1992
Cultural Algorithm (CA) [21]	Human cultural evolution process	1994
Differential Evolution (DE) [22]	Darwin's theory of evolution	1997
Grammatical Evolution (GE) [23]	Biological evolutionary process	1998
Imperialist Competitive Algorithm (ICA) [24]	The idea of imperialism	2007
Differential Search Algorithm (DSA) [25]	Migration of superorganisms	2011
Backtracking Search Algorithm (BSA) [26]	Evolutionary concepts with memory	2013
Stochastic Fractal Search (SFS) [27]	Random growth in nature and the use of fractal mathematical concepts	2014
Synergistic Fibroblast Optimization (SFO) [28]	The fibroblast in the skin wound healing process	2018
Wildebeests Herd Optimization (WHO) [29]	Wildebeest herding behavior	2019
Learner Performance based Behavior (LPB) [30]	Distribution of students in different departments at a university	2021

Table 2
A brief review of swarm intelligence-based algorithms.

Algorithm	Inspiration	Year
Ant Colony Optimization (ACO) [31]	Ant colony	1992
Particle Swarm Algorithm (PSO) [32]	Social behavior of bird swarms	1995
Marriage in honey Bees Optimization (MBO) [33]	The mating and fertilization process of honey bees	2001
Artificial Fish Swarm Algorithm (AFSA) [34]	Collective intelligence of fish swarms	2002
Bacterial Foraging algorithm (BFA) [35]	Social foraging and chemotactic behavior of bacteria	2002
Bee Swarm Optimization (BSO) [36]	Foraging behavior of honey bees	2003
Glowworm Swarm Optimization (GSO) [37]	Glow behavior of glowworms	2005
Cat Swarm Optimization (CSO) [38]	Natural behavior of cats	2006
Artificial Bee Colony (ABC) algorithm [39]	Foraging behavior of honey bees	2007
Roach Infestation Optimization (RIO) [40]	Cockroaches' behavior	2008
Cuckoo Search (CS) [41]	The brood parasitism of some cuckoo species	2009
Group Search Optimizer (GSO) [42]	Animal searching behavior	2009
Bat Algorithm (BA) [43]	Echolocation behavior of bats	2010
Firefly Algorithm (FA) [44]	Social behavior of fireflies	2010
Termite Colony Optimization (TCO) [45]	Termite behavior	2010
Fruit Fly Optimization (FOA) [46]	Fruit foraging behavior of fruit fly	2011
Flower Pollination Algorithm (FPA) [47]	Pollination process of flower species	2012
Krill Herd (KH) [48]	Herding behavior of krill individuals	2012
Dolphin Echolocation (DE) [49]	Echolocation ability of dolphins	2013
Swallow Swarm Optimization (SSO) [50]	The behaviors of swallows	2013
Animal Migration Optimization (AMO) [51]	The behavior of animal migration	2014
Ant Lion Optimizer (ALO) [52]	Hunting behavior of ant lions	2015
Artificial Algae Algorithm (AAA) [53]	Living behavior of micro algae	2015
Dragonfly Algorithm (DA) [54]	Dragonfly swarming behavior	2015
Virus Colony Search (VCS) [55]	Virus infection and spread strategies	2015
Crow Search Algorithm (CSA) [56]	Intelligent behavior of crows	2016
Dolphin Swarm Algorithm (DSA) [57]	Dolphins' mechanisms in catching sardine swarms	2016
Whale Optimization Algorithm (WOA) [58]	Social behavior of humpback whales	2016
Grasshopper Optimization Algorithm (GOA) [59]	Swarming behavior of grasshoppers	2017
Mouth Brooding Fish (MBF) algorithm [2]	Fish spawning behavior	2017
Salp Swarm Algorithm (SSA) [10]	Foraging behavior of salps	2017
Spotted Hyena Optimizer (SHO) [60]	Social behavior of spotted hyenas	2017
Squirrel Search Algorithm (SSA) [11]	Locomotion behaviors of squirrels	2018
Harris Hawks Optimizer (HHO) [61]	Cooperative behavior and stalking style of Harris's hawks	2019
Sparrow Search Algorithm (SSA) [9]	Foraging behavior of sparrows	2020
Capuchin Search Algorithm (CapSA) [6]	Hunting behavior of capuchins	2021
Chameleon Swarm Algorithm (CSA) [4]	Hunting behavior of chameleons	2021
Aquila Optimizer(AO) [7]	Foraging behavior of Aquila	2021

attention from both academia and industry. More narrowly, they have been disciplined to tackle real-life problems in a wide range of scientific and engineering disciplines. Examples of these fields include, but are not limited to, image processing [97], dynamic economic dispatch problems [98], the realm of process control [99], variable length wireless sensor network [100], heat and power dynamic economic dispatch problems [101] and many other areas [3,102].

This broad acceptance of meta-heuristics is relevant to that they are quite flexible and do not depend on the nature of the problem of interest, do not demand gradient information, and have been shown to be successful in escaping local minimums when the optimization problems to be solved involve many local solutions [3]. The flexibility of meta-heuristics entitles them, as suitable tools, to efficiently solve problems without making utmost modifications to the structure of the algorithms. It is noteworthy that the first and second merits stem from the fact that meta-heuristics address and solve problems by accepting them as black boxes as they only request knowledge of the problems' input and output parameters. The computation of the derivative of the search space of the problems of concern is not required, because meta-heuristics reside on a family of stochastic approaches, where they take trait of stochastic operators. This feature has been widely emphasized, where they have proven

Table 3
A brief review of human-based algorithms.

Algorithm	Inspiration	Year
Society Civilization Algorithm (SCA) [62]	Leadership phenomena of humans	2003
Seeker Optimization Algorithm (SOA) [63]	The action of human randomized search	2006
Imperialist Competitive Algorithm (ICA) [64]	Imperialistic competition	2007
Human-Inspired Algorithm (HIA) [65]	People's intelligence	2009
League Championship Algorithm (LCA) [66]	Championship process in sport leagues	2009
Social Emotional Optimization Algorithm (SEOA) [67]	Human social behaviors	2010
Brain Storm Optimization (BSO) [68]	Brainstorming process	2011
Teaching-Learning-Based Optimization (TLBO) [69]	Teaching and learning in a classroom	2011
Anarchic Society Optimization (ASO) [70]	A social group behaving in a chaotic way to improve its situation	2012
Human Mental Search (HMS) [71]	Exploration strategies of the bid space in online auctions	2017
Volleyball Premier League (VPL) [72]	Simulation of some phenomena in volleyball	2018
Gaining Sharing Knowledge based Algorithm (GSK) [1]	Acquisition and exchange of knowledge during a person's lifespan	2020
Coronavirus Herd Immunity Optimizer (CHIO) [73]	Herd immunity concept to respond to COVID-19	2021
Ali Baba and the Forty Thieves (AFT) [14]	The tale of Ali Baba and the forty thieves	2021

Table 4
A brief review of physics-based algorithms.

Algorithm	Inspiration	Year
Simulated Annealing (SA) [75]	Annealing process in metallurgy	1983
Variable Neighborhood Search (VNS) [76]	The idea of neighborhood change	1995
Big Bang-Big Crunch (BB-BC) [77]	Evolution of universe	2005
Central Force Optimization (CFO) [78]	Gravitational kinematics metaphor	2007
Intelligent Water Drops (IWD) [79]	Water drops in a river	2007
Slime Mold Algorithm (SMA) [80]	The oscillation of slime mold	2008
Gravitational Search Algorithm (GSA) [81]	Gravity law and mass interactions	2009
Charged System Search (CSS) [82]	Principles from physics and mechanics	2010
Electro-Magnetism Optimization (EMO) [83]	Electromagnetism principles	2011
Black Hole Algorithm (BHA) [84]	Black hole in general relativity and cosmology	2012
Water Cycle Algorithm (WCA) [85]	Water cycle process	2012
Mine Blast Algorithm (MBA) [86]	Mine bomb explosion	2013
Colliding Bodies Optimization (CBO) [87]	Collision of two things in one dimension	2014
Lightning Search Algorithm (LSA) [88]	Natural phenomenon of lightning	2015
Weighted Superposition Attraction (WSA) [89]	Superposition precept	2015
Multi-Verse Optimizer (MVO) [90]	Multi-verse theory	2016
Thermal Exchange Optimization (TEO) [91]	Newton's law of cooling	2017
Find-Fix-Finish-Exploit-Analyze (F3EA) [92]	Selecting objects to be destroyed in warfare	2018
Henry Gas Solubility Optimization (HGSO) [93]	The huddling behavior of gas	2019
Equilibrium Optimizer (EO) [74]	Physics-based source and sink models	2020
Archimedes Optimization Algorithm (AOA) [94]	Archimedes' principle of physics	2021
Lichtenberg Algorithm (LA) [95]	Lichtenberg figure patterns	2021

their success in moving away from local minimums when addressing real problems, which often possess a large number of local minimums [11]. Finally, simplicity and ease of use are other momentous benefits as most of these methods imitate simple rules and behaviors in flocks or swarms.

1.3. Motives and contributions of the proposed work

A more widespread method to judge the adequacy of a new meta-heuristic is to show its competitiveness opposed to mathematical programming methods and other meta-heuristics in solving optimization problems. Whilst existing meta-heuristics have proven their usefulness in reliably finding the global optima during optimization, it is not guaranteed that every meta-heuristic can effectively find the global optimal for all kinds of problems. This is explicitly stated in the '"no-free-lunch"' (NFL) theory [103], where there is no generic optimization method that can arrive at the optimal solution for all types of problems. In other words, if a meta-heuristic method is finely tuned to get high accuracy in one class of problems or even some methods in the same class, then it will be offset by performance in another class of problems or other methods in the same class. In this regard, the NFL theory keeps this field of research open and pushes researchers to devise new methods to reinforce optimization and raise accuracy to address complex real-world problems that are consistently emerging in the world on account of high-tech advances [53]. In essence, there is no existing meta-heuristic that mathematically models and implements the dynamic behaviors of great white sharks in the depth of ocean that focuses on the exceptional sense of hearing and smell during hunting for prey. The above reasons are the main drivers behind this work.

This paper presents a new optimization algorithm named White Shark Optimizer (WSO), which is inspired by the scholastic behavior of white sharks while foraging in nature to successfully survive in the deep ocean. This algorithm was proposed in the hope of solving both constrained and unconstrained real-world optimization problems that are hard to tackle with existing optimization algorithms. Another expectation of WSO is to find better solutions than the existing ones. Since exploration and exploitation are the two main features for the prosperity of any meta-heuristic, they are effectively designed in WSO to strike a proper balance between them. The key contributions of this work can be briefly summed up as follows: (1) a new meta-heuristic that mimics the behaviors of white sharks' school and their sense of hearing and smell is presented, (2) the performance of the proposed WSO was assessed on 29 widely known benchmark functions that belong to the CEC-2017 test suite, in which an overall comparison was made with other meta-heuristics, and (3) the applicability of WSO was further examined in solving the CEC-2011 benchmark problems, and the obtained results were compared with other meta-heuristics.

1.4. Advantages of WSO for optimization problems

The proposed WSO has several advantages for global optimization problems, such as its expected flexibility to deal with different types of optimization problems, where many types of problems demand a degree of flexibility beyond that which is possible with WSO with only a few parameters to be tuned as presented later in this work. The proposed mathematical model for WSO makes it relevant to address various kinds of engineering optimization problems, especially those of high dimensionality. It is anticipated that the simplicity and robustness of WSO make it rapid and accurate to find the global solution for difficult optimization problems with high convergence speed as a third advantage. A fourth advantage of WSO for global optimization is to be a potent candidate with a broad interest in developing low-cost and powerful solutions to challenge real-world optimization problems.

The rest of this paper is formulated as follows: In Section 2, we introduce the key inspiration notions along with the basic ideas of the algorithm presented in this work. Section 3 then presents in detail the mathematical formulation and implementation of the proposed algorithm. The evaluation, convergence and statistical test results of WSO and other algorithms in various classes of benchmark functions are presented in Section 4. Finally, the main conclusions and future trends of this work are given in Section 5.

2. Inspiration

The meta-heuristic algorithm presented in this paper is based on the dynamic behavior of white sharks. We have found in this species of shark several intrinsic traits that inspired us to develop the proposed WSO. White sharks, also known as white pointers or great white sharks, are some of the strongest and most hazardous predacious sharks in the world [104]. White sharks are highly acclimatized predators and stunning hunters, armed with powerful muscles, sturdy eyesight for well-contrasted vision, and a keen sense of smell. Their gigantic jaws are lined with up to 300 acutely pointed, vulgarly serrated, triangular teeth arranged in several rows. Their prey includes other sharks, crustaceans, mollusks, seabirds, small whales, dolphins, squids, turtles, seals, sea lions, walruses, sea turtles, porpoises, and sometimes penguins. They customarily catch prey through an ambuscade, where a white shark tries to rush a prey by surprise and strikes an abrupt and large fatal bite. White sharks are streamlined, torpedo-shaped swimmers with mighty tails that can impel them in the water. They swim toward prey with undulating motion, and can even leave the water completely, and explode like whales when attacking prey from underneath [104]. The most interesting facts of the collective conduct of great white sharks are their way of catching prey using their method of swimming as well as their uncommon senses of hearing and smelling the scent of prey.

2.1. Tracking the prey

Similar to any organism that lives in nature, white sharks wander the ocean while hunting for prey, so they change their position accordingly. In this respect, they use almost every method at their disposal to track down, stalk and locate their prey. They have a bunch of senses as it is perceived in Fig. 1 that are incorporated and complementary.

First, great white sharks have a surprisingly effective sense of hearing, which they use to scout a large space while searching for prey. Second, they have an acute sense of smell to smell the scent of prey. These features help them to explore the entire space and exploit each possible area of the search domain to hunt down prey.

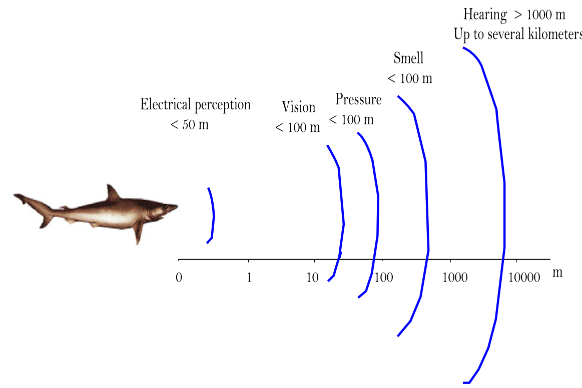


Fig. 1. White shark's senses: smell, sight and hearing.

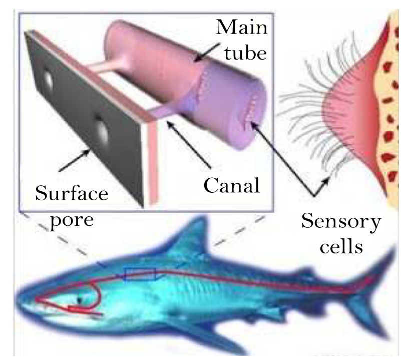


Fig. 2. A great white shark with a hearing line sensor shown on its torso.

2.2. Search for prey (exploration)

Great white sharks explore the realm of search space while searching for prey using unfamiliar sense of hearing. They can hear from the entire length of their bodies, in which there are two lines running on either side of their bodies as clearly shown in Fig. 2 [105].

These two lines can detect changes in water pressure, revealing the movements of prey. The changes in water pressure emitted by a turbulent prey will entice the attention of white sharks to move towards the prey. They even have organs that can sense the teeny electromagnetic fields produced during the movement of prey. Then, based on the frequency of waves drifting to them during prey's motion and its turbulence, they can precisely locate the position of prey along with its size. When a white shark is so close to its target, it will be able to pick up the electromagnetic fields, and when it locates a prey, it will move to the prey with an undulating motion that can be illustrated as shown in Fig. 3.

The wavy velocity of great white sharks can be described by the following mathematical formula:

$$v = xf \tag{1}$$

where v is the speed of the wavy motion, x is the wavelength that defines the distance a white shark travels in an undulating motion to complete one complete revolution, and f represents the wave frequency of the wavy motion which is determined by the number of revolutions (i.e., cycles) that the white shark completes per second, where cycle per second is known as Hertz (Hz).

2.3. Search for prey (exploitation)

Great white sharks exploit every possible area in the space domain to spot prey using their exceptional sense of smell. When

(a) (b)

Fig. 3. Wavy motion behavior of great white sharks: (a) a great white shark rippling to the left, (b) a great white shark rippling to the right.

a white shark approaches its prey, its sense of smell begins to act [105]. Remarkably, when great white sharks are close to their prey, their sense of smell can grow in an exponential manner until they accurately pinpoint the likely position of prey. To update the position of white sharks as they move to a prey, the following equation of motion with constant acceleration can be used:

$$x = x_i + v_i \Delta t + \frac{1}{2} a\,(\Delta t)^2 \tag{2}$$

where x denotes the new position of the white shark, x_i is the primitive position, v_i is the initial velocity of the white shark, Δt represents the time interval between the initial and current positions and a is the acceleration factor which is constant.

In many cases, prey like seals leave their scent after they leave off their place, so great white sharks find no prey there when they are too close to that scent. In this case, they have to randomly search in nearby locations and randomly explore other areas in the search space utilizing their active senses of sight, hearing, and smell.

In this work, the overall picture of the hunting behavior of great white sharks and their effective ways of trailing and stalking prey have guided us to the mathematical models evolved for WSO and carry out optimization. Below is a thorough characterization of these models and the proposed WSO.

3. White Shark Optimizer (WSO)

This section describes in detail the mathematical models of the proposed WSO, which were developed to characterize the behavior of white sharks while foraging. This includes hunting for and tracking prey.

Great white shark has the aptitude to locate the location of prey (i.e., a food source) in deep ocean. However, there is no notion about the location of the food source in a particular search space. In this context, white sharks have to search extensively to locate food sources in the depths of the ocean. In this work, three behaviors of great white sharks were used to find prey (i.e., the optimal food source), which are: (1) the movement towards prey based on the hesitation of the waves that occurs due to movement of prey. In this, the white shark uses an undulating motion to navigate to prey by utilizing its associated senses of hearing and smell, (2) the random search for prey in the depths of the ocean. For this, great white sharks move towards the location of prey and stay close to the optimal prey, and (3) the behavior of white shark in locating nearby prey. In this, great white shark uses fish school behavior and moves towards the best white shark that is very close to the optimal prey. Based on these behaviors, the locations of all white sharks will be updated about the best optimal solutions in case the prey is not found appropriately. These behaviors are mathematically modeled as shown below.

3.1. Initialization of WSO

As WSO is a population-based algorithm, it commences by randomly generating a pool of initial solutions when initiating the optimization process used to solve an optimization problem. A population of n white sharks (i.e., population size), in a d-dimensional search space (i.e., dimension of the problem), with the position of each white shark, indicates a candidate solution to a problem can be described in a $2d$ matrix as shown below:

$$w = \begin{bmatrix} w_1^1 & w_2^1 & \dots & \dots & w_d^1 \\ w_1^2 & w_2^2 & \dots & \dots & w_d^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ w_1^n & w_2^n & \dots & \dots & w_d^n \end{bmatrix} \tag{3}$$

where w stands for the position of all white sharks in the search space, d denotes the number of decision variables for a given problem and w_d^i indicates the location of the ith white shark in the dth dimension.

The initial population is created in the search domain with a uniform random initialization as given below:

$$w_j^i = l_j + r \times (u_j - l_j) \tag{4}$$

where w_j^i is the initial vector of the ith white shark in the jth dimension, u_j and l_j represent the upper and lower bounds of the search space in the jth dimension, respectively, and r is a random number created in the interval [0, 1].

The quality of each candidate solution for each new location of a white shark is assessed on the basis of a fitness function defined for that purpose. Then the current position is renovated if the new position is better than the current one. In the simulation of WSO, the white shark remains at its position if it is better than the new position.

3.2. Movement speed towards prey

As white sharks are creatures with a penchant for survival, they spend most of their time hunting and tracking prey. They generally use every method to stalk and track prey using their extraordinary senses such as hearing, sight and smell. When a white shark perceives the location of a prey based on the hesitation of the waves it hears as the prey is moving, it moves to the prey in an undulating motion that can be defined as shown in Eq. (6).

$$v_{k+1}^i = \mu \left[v_k^i + p_1 (w_{gbest_k} - w_k^i) \times c_1 + p_2 (w_{best}^{v_k^i} - w_k^i) \times c_2 \right] \tag{5}$$

where $i = 1, 2, \dots, n$, is the white shark's index for a population of size n, v_{k+1}^i denotes the new velocity vector of the ith white shark in $(k+1)$th step, v_k^i defines the current speed vector of the ith white shark in kth step, w_{gbest_k} represents the global best position vector obtained so far by any white shark in the kth iteration, w_k^i is the current position vector of the ith white shark in kth step, $w_{best}^{v^i}$ is the ith best known position vector known to the swarm and v^i is the ith index vector of the white sharks reaching the best position defined as shown in Eq. (6), c_1 and c_2 are two uniformly created random numbers in the range [0, 1], p_1 and p_2 represent the forces of the white sharks that control the effect of w_{gbest_k} and $w_{best}^{v^i}$ on w_k^i, respectively, which are computed as given in Eqs. (7) and (8), respectively, and μ represents the constriction factor suggested in WSO to analyze the convergence behavior of white sharks which is defined as presented in Eq. (9).

$$v = \lfloor n \times rand\,(1, n) \rfloor + 1 \tag{6}$$

where $rand(1, n)$ is a vector of random numbers generated with a uniform distribution in the range [0, 1].

$$p_1 = p_{max} + (p_{max} - p_{min}) \times e^{-(4k/K)^2} \tag{7}$$

$$p_2 = p_{min} + (p_{max} - p_{min}) \times e^{-(4k/K)^2} \tag{8}$$

where k and K stand for the current and maximum number of iterations, respectively, p_{min} and p_{max} represent the initial and subordinate velocities to achieve good motion for white sharks. The values of p_{min} and p_{max} after rigorous analysis were found to be as 0.5 and 1.5, respectively.

$$\mu = \frac{2}{\left|2 - \tau - \sqrt{\tau^2 - 4\tau}\right|} \tag{9}$$

where τ denotes the acceleration coefficient which is equal to 4.125 where this value was found after thorough analysis.

3.3. Movement towards optimal prey

Great white sharks outlay most of their time searching for potential prey, where an optimal or suboptimal prey can be located. Accordingly, the positions of the white sharks are constantly changing. They typically move toward prey when they either hear the waves caused by the movement of prey or smell the scent of prey. In some cases, the prey leaves its location either because a white shark moves to it, or to search for food. Often, the prey leaves its scent in that position, where the white shark can still smell the prey. In this case, the white shark navigates in random locations in search of prey as the case of the behavior of fish school looking for a food source. In this context, the position updating strategy defined in Eq. (10) was used to describe the behavior of white sharks as they move towards prey.

$$w_{k+1}^i = \begin{cases} w_k^i \cdot \neg \oplus w_o + u \cdot a + l \cdot b; & rand < mv \\ w_k^i + v_k^i / f; & rand \geq mv \end{cases} \tag{10}$$

where w_{k+1}^i refers to the new position vector of the ith white shark in the $(k+1)$th iteration step, $\neg$ is a negation operator, a and b are one-dimensional binary vectors defined as given by Eqs. (11) and (12), respectively, l and u denote the lower and upper limits of the search space, respectively, w_o denotes a logical vector defined as shown in Eq. (13), f represent the frequency of the wavy motion of a white shark defined as presented in Eq. (14), *rand* defines a random number created in the range from 0 to 1 and mv represents the movement force that increases, with the number of iterations, as the white shark approaches prey, which is defined as expressed in Eq. (15).

$$a = \mathrm{sgn}(w_k^i - u) > 0 \tag{11}$$

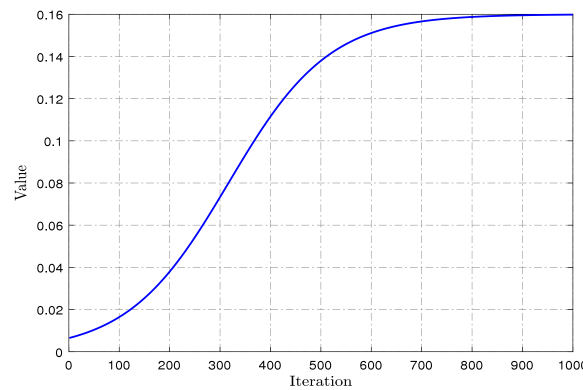


Fig. 4. A function of iterations that represents the general growth trend sense of hearing and smell of white sharks when approaching a prey.

$$b = \mathrm{sgn}(w_k^i - l) < 0 \tag{12}$$

$$w_o = \oplus(a, b) \tag{13}$$

where $\oplus$ is a bit-wise xor operation.

Eqs. (11) and (12) are important to support solutions to arbitrarily behave in the search space and are essential to assist white sharks to explore all potential areas of the search space.

$$f = f_{min} + \frac{f_{max} - f_{min}}{f_{max} + f_{min}} \tag{14}$$

where f_{min} and f_{max} denote the minimum and maximum frequencies of the undulating motion, respectively, and *rand* represents a random number uniformly dispensed within the scope [0, 1]. The values of f_{min} and f_{max} for the problems handled in this work are 0.07 and 0.75, respectively. These values were chosen after precise analysis and tested for a wide range of problems, but can be well adapted for other problems as appropriate.

$$mv = \frac{1}{\left(a_0 + e^{(K/2 - k)/a_1}\right)} \tag{15}$$

where a_0 and a_1 are two positive constants employed to manage exploration and exploitation behaviors.

The parameter mv was proposed to express the strength of the white shark's sense of hearing and smell, which increases as a function of iterations. This function is drawn over the course of iterations as displayed in Fig. 4.

Fig. 4 shows the schematic diagram of the parameter mv and its influence on the search ability. Small values of mv results in local search (at the vicinity of w_{k+1}^i), which elicits white sharks to search in smaller areas. Large values of mv lead to global search (far from w_{k+1}^i), which prompts white sharks to explore larger and farther areas in the search space. Basically, this parameter is principally important to augment the possibility of achieving a proper balance between exploration and exploitation.

The parameter mv in Eq. (15) was proposed as a function of time to assure convergence by accelerating the search speed as well as strengthening exploration and exploitation features of the proposed algorithm. From a mathematical point of view, large values of mv prohibit more searches and small values of mv foster the search intensity in the search space. Thence, the values of mv can control the search intensity of white sharks to efficiently locate prey. Therefore, it is essential to find the suitable values of the coefficients a_0 and a_1 for the function mv. Their values for all of the problems then addressed in this work are 6.25 and 100, respectively. They were obtained after deep analysis and extensive experimental testing on a large number of optimization problems.

The first part of Eq. (10), precisely when $rand < mv$, describes that white sharks randomly update their position around the prey (i.e., food source) in a given search space. This is especially performed when a prey leaves the place and leaves its scent in that place, and then the white shark randomly follows the scent around the prey. In this way, white sharks can take advantage of their good senses of hearing and smell to exploit every area of the search space. This part mainly shows that white sharks randomly change their position in many directions and areas while hunting for optimal prey. The second part of Eq. (10), is expressed in this way based on the second equation of motion with constant acceleration. This formula is roughly similar to what is shown in Eq. (2). However, the time was omitted from this formula because time is iteration in optimization problems, where the discrepancy between iterations is always equal to 1. This part simulates that white sharks normally move to a prey, particularly when they hear the drifting of waves caused by a prey's motion. This case is simulated when $rand \geq mv$. This part chiefly spells out that white sharks can update their position in the search space as per they hear prey.

3.4. Movement towards the best white shark

Great white sharks can maintain their position towards the best one that is close to prey. This behavior is formulated as shown in Eq. (16).

$$\acute{w}_{k+1}^{i} = w_{gbest_k} + r_1 \vec{D}_w \mathrm{sgn}(r_2 - 0.5) \quad r_3 < s_s \tag{16}$$

where $\acute{w}_{k+1}^{i}$ is the updated position of the ith white shark with respect to the position of prey, $\mathrm{sgn}(r_2 - 0.5)$ gives either 1 or -1 to change the direction of the search, the variables r_1, r_2 and r_3 are random numbers that lie in the range of [0, 1], $\vec{D}_w$ is the distance between prey (i.e., food source) and white shark, defined as given in Eq. (17) and s_s is a parameter suggested to express the strength of senses of smell and sight for white sharks when they follow other white sharks that are close to optimal prey, which is defined as given in Eq. (18).

$$\vec{D}_w = \left| rand \times \left(w_{gbest_k} - w_k^i \right) \right| \tag{17}$$

where $rand$ is a random number in range of [0, 1] and w_k^i is the current position of white shark in respect to w_{gbest_k}.

$$s_s = \left| 1 - e^{(-a_2 \times k/K)} \right| \tag{18}$$

where a_2 is a positive constant utilized to control exploration and exploitation behaviors. The value of a_2 is 0.0005, for all of the test problems addressed in this study. This value was chosen after meticulous assessment for a large number of problems, which can be tuned for other problems on demand.

3.5. Fish school behavior

In order to mathematically simulate the behavior of the school of white sharks, the first two optimum best solutions were preserved and the position of other white sharks was updated in accordance to these optimal positions. The formula given below was proposed to define the fish school behavior of white sharks:

$$w_{k+1}^{i} = \frac{w_k^i + \acute{w}_{k+1}^{i}}{2 \times rand} \tag{19}$$

where $rand$ represents a random number uniformly distributed within the interval [0, 1].

Eq. (18) demonstrates that white sharks can update their position in compliance with the position of the best white shark that has reached the best position which is very close to prey.

The final position of great white sharks (i.e., search agents) would be somewhere within the search space that is very close to optimal prey. Fish school behavior and the movement of white sharks towards the best white shark identify the collective conduct of WSO and this swells the scope for better exploration and exploitation features.

3.6. Exploration ability of WSO

There are several parameters used to heighten the exploration ability of WSO which are explained as follows:

- μ: controls the exploration behavior of WSO which also helps to avert early convergence and precludes solutions from descending into local optima. The value of μ depends on the parameter τ, whereby fine-tuning of this parameter boosts the global search ability towards the global optimal solution.
- p_1 and p_2: these parameters, as functions of iterations, control the velocity update of white sharks and achieve a stable balance between exploration and exploitation during WSO's global and local searches. They also control the influence of the best position known to the entire fish school of white sharks on the current position achieved. Thus, to provide reasonable exploration ability with precise solutions, these parameters are variable during the course of iterations of WSO. .
- mv: this parameter was selected on the basis of empirical tests, where it should not be greater than 1.0, as this would devalue the exploration performance. In the initial iterations, white sharks are all long way away from prey. Updating the position of white sharks based on Eq. (10) with the utilization of mv helps WSO to globally and locally search the space. The higher the coefficient a_1 in mv, the better the exploration capacity and hence the lower exploitation accuracy. Likewise, the lower the a_0, the better the exploitation capacity and the lower the exploration capacity. For this reason, this parameter was sufficiently well-tuned to expand both the exploration and exploitation abilities. It is found, based on experimental testing, that values largely more than 20 for a_0 and largely more than 100 for a_1 encourage search agents to search on boundaries.
- $\mathrm{sgn}(r_2 - 0.5)$ in Eq. (16): controls exploration's direction, and since r_2 is located in [0, 1] with a consistent distribution, there is an equal probability of negative and positive signs.
- r_1 in Eq. (16): assists the solutions to behave randomly in a given search space and a variation of this variable can provide better exploration.

3.7. Exploitation ability of WSO

The key parameters to conduct exploitation and local search in WSO are explained as follows:

- μ: this parameter is used to intensify the amount of exploitation of WSO as well. The value of μ relies on the value of the parameter τ, where tuning this parameter diminishes the probability of slack in local optima solutions.
- p_1 and p_2: these parameters have an impact on the success of WSO, which is attributed to their actions in balancing global and local search behaviors.

- mv: by the growth of iterations, exploration goes away and exploitation goes in. Thus, in the final iterations, where a white shark is in close proximity to a prey, mv will assist WSO in a local search around the prey, leading to exploitation. The coefficients a_0 and a_1 in mv control the exploitation feature as well. They determine the amount of exploitation by sculpting around the best solution.
- $\mathrm{sgn}(r_2 - 0.5)$ in Eq. (16): controls the exploitation by designating the direction of the local search, as there is a similar possibility of negative and positive signs since r_2 is located in [0, 1].
- r_1 in Eq. (16): helps to increase the randomness of solutions in the search space and so a sensible exploitation process.

3.8. Complexity analysis

The computational complexity of an algorithm can be described by a function that connects the problem's input size to the algorithm's run-time. To this end, Big-O notation is employed here as a familiar expression to put forward the computational complexities of time and space of the proposed WSO as given below.

3.8.1. Time complexity

Time complexity issue relies on the number of white sharks (n), the dimensions of the given problem (d), the number of iterations (K) and the function assessment's cost (c). In specific terms, the time complexity of WSO can be rendered as:

$$\begin{aligned}\mathcal{O}(WSO) &= \mathcal{O}(problem\ Def.) + \mathcal{O}(init.) \\ &\quad + \mathcal{O}(cost\ function) + \mathcal{O}(Sol.\ update)\end{aligned} \tag{20}$$

where the time complexities of the components of Eq. (21) can be defined as follows:

1. Initialization of problem definition demands $\mathcal{O}(1)$ time.
2. Initialization of population creation demands $\mathcal{O}(n \times d)$ time.
3. Assessment of the cost function demands $\mathcal{O}(K \times c \times n)$ time.
4. Evaluation of the solution update demands $\mathcal{O}(K \times n \times d)$ time.

Thus, the general time complexity of WSO can be expressed as follows:

$$\mathcal{O}(WSO) = \mathcal{O}(1 + nd + Kcn + Knd) \tag{21}$$

As $1 \ll Kcn$, $1 \ll Knd$, $nd \ll Kcn$ and $nd \ll Knd$, Eq. (21) can be reduced to Eq. (22):

$$\mathcal{O}(WSO) \cong O(Kcn + Knd) \tag{22}$$

As it is revealed, the time complexity issue of WSO is of the polynomial order. In this, the proposed WSO can be considered as a computational efficient optimization algorithm.

Briefly, it can be seen from Eq. (22) that the main factors for determining the computational complexity of WSO in solving an optimization problem can be defined as: the number of decision variables of the problem (i.e., d), the cost of the objective function of the problem (i.e., c), the number of search agents (i.e., n) and the number of iterations (i.e., K), both of which depend on the nature of the problem.

3.8.2. Space complexity

The space complexity of WSO in relation to the amount of memory space relies on the parameters of both the number of white sharks and the dimensions of the problem. This identifies how much space the proposed WSO will require during the initialization process. In view of this, the space complexity of WSO can be stated as follows:

$$\mathcal{O}(nd) \tag{23}$$

Algorithm 1 A pseudo code summarizing the iterative optimization process of WSO.

Initialize the parameters of the problem
Initialize the parameters of WSO
Randomly generate the initial positions of WSO
Initialize the velocity of the initial population
Evaluate the position of the initial population
while ($k < K$) **do**
  Update the parameters ν, p_1, p_2, μ, a, b, w_o, f, mv and s_s using Eqs. (6), (7), (8), (9), (11), (12), (13), (14), (15) and (18), respectively.
  for i=1 to n **do**
    $v_{k+1}^i = \mu\left[v_k^i + p_1(w_{gbest_k} - w_k^i) \times c_1 + p_2(w_{best}^{\nu^i{}_k} - w_k^i) \times c_2\right]$
  end for
  for i=1 to n **do**
    if $rand < mv$ **then**
      $w_{k+1}^i = w_k^i \cdot \neg \oplus w_o + u \cdot a + l \cdot b$
    else
      $w_{k+1}^i = w_k^i + v_k^i / f$
    end if
  end for
  for i=1 to n **do**
    if $rand \leq s_s$ **then**
      $\vec{D}_w = \left|rand \times \left(w_{gbest_k} - w_k^i\right)\right|$
      if $i == 1$ **then**
        $w_{k+1}^i = w_{gbest_k} + r_1 \vec{D}_w \mathrm{sgn}(r_2 - 0.5)$
      else
        $\acute{w}_{k+1}^i = w_{gbest_k} + r_1 \vec{D}_w \mathrm{sgn}(r_2 - 0.5)$
        $w_{k+1}^i = \frac{w_k^i + \acute{w}_{k+1}^i}{2 \times rand}$
      end if
    end if
  end for
  Adjust the position of the white sharks that proceed beyond the boundary
  Evaluate and update the new positions
  $k = k + 1$
end while
Return the optimal solution obtained so far

3.9. Implementation and analysis of WSO

Great white sharks always try to move ahead towards the global optimum solution (i.e., prey) during the route of function evaluations. This is because the white sharks are very probably to locate a better solution by exploring and exploiting the surrounding area in the search space. This ability is implemented based on the position of the best white sharks and prey. Therefore, white sharks are continually able to explore and exploit the promising areas where prey can be there. The pseudo code of WSO can be summed up by the key steps presented in Algorithm 1.

Algorithm 1 presents that WSO launches the optimization process by generating the position of white sharks at random. The position of white sharks is updated at each function evaluation using Eq. (10). If any of them passes outside of the search space, it will be returned to the search area based on the simulated steps of WSO. The fitness function is used to evaluate the solutions, whereby the fitness criterion is recomputed within each function evaluation to locate the white shark with the fittest value. It is indicated that the most appropriate solution is the best position for a white shark that finds prey. All the steps of Algorithm 1 excluding the initialization steps are reiterated at each function evaluation until the specified maximum number of function evaluations is achieved. The theoretical claims of WSO described

Table 5
Parameter settings of WSO and other search algorithms.

Algorithm	Specifications	Population size
WSO	$f_{min} = 0.07$, $f_{max} = 0.75$ $\tau = 4.125$, $a_0 = 6.25$, $a_1 = 100$, $a_2 = 0.0005$	100
TLBO	No especial parameters	50 because this algorithm has two main stages
SFS	Maximum diffusion number is set to 1	50 because this algorithm has two main stages
DE	F = 0.5, CR = 0.9	100
GA	Roulette wheel selection, mutation probability = 0.01, single point crossover with a probability of 1.	100
GSK	P = 0.1, $k_f = 0.5$, $k_r = 0.9$, K = 10	100
AMO	The number of animals in each set is 5	50, the population size is 50 because of this algorithm has two phases.
PSO	$\omega = 0.6$, $c_1 = 2$ and $c_2 = 2$	100
BBO	Habitat modification probability = 1, immigration probability bounds for each gene = [0, 1], mutation probability = 0, maximum migration and immigration rates for each island = 1, and the step size for numerical integration of probabilities = 1.	100
ACO	Pheromone update constant = 20, initial pheromone value = 1E−06, exploration constant = 1, local and global pheromone decay rates = 0.5 and 0.9, respectively, and visibility and pheromone sensitivities = 5 and 1, respectively.	100

Table 6
Statistical results of WSO over 51 independent runs on the CEC-2017 functions of 10 variables with 100,000 FEs.

F	Best	Median	Ave	Worst	Std
C-17-f1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
C-17-f3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
C-17-f4	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
C-17-f5	2.98E+00	7.95E+00	8.52E+00	14.92E+00	3.21E+00
C-17-f6	0.00E+00	2.12E−08	1.40E−07	4.04E−06	5.67E−07
C-17-f7	1.28E+01	1.65E+01	1.79E+01	3.60E+01	4.60E+00
C-17-f8	9.94E−01	7.95E+00	8.72E+00	2.78E+01	4.92E+00
C-17-f9	0.00E+00	0.00E+00	1.76E−02	4.54E−01	7.02E−02
C-17-f10	1.18E+02	4.70E+02	5.44E+02	1.17E+03	2.84E+02
C-17-f11	0.00E+00	2.98E+00	4.19E+00	1.29E+01	3.49E+00
C-17-f12	0.00E+00	3.75E+02	3.66E+02	7.18E+02	1.77E+02
C-17-f13	0.00E+00	1.72E+01	2.34E+01	8.33E+01	1.93E+01
C-17-f14	9.94E−01	2.39E+01	2.15E+01	4.58E+01	1.28E+01
C-17-f15	1.65E−02	5.25E+00	6.57E+00	2.60E+01	5.39E+00
C-17-f16	4.86E−02	2.13E+01	1.27E+01	1.22E+02	3.57E+01
C-17-f17	2.36E+00	2.63E+01	2.52E+01	4.97E+01	1.25E+01
C-17-f18	1.00E+00	2.21E+01	2.15E+01	4.53E+01	1.12E+01
C-17-f19	3.09E−02	2.54E+00	2.73E+00	1.44E+01	2.24E+00
C-17-f20	9.95E−01	2.42E+01	2.43E+01	5.31E+01	1.13E+00
C-17-f21	1.00E+02	2.04E+02	1.68E+02	3.88E+02	5.31E+01
C-17-f22	0.00E+00	1.00E+02	9.40E+01	5.23E+02	2.47E+01
C-17-f23	3.02E+02	3.10E+02	3.11E+02	3.52E+02	5.34E+00
C-17-f24	1.00E+02	3.34E+02	2.88E+02	3.96E+02	9.43E+01
C-17-f25	3.97E+02	3.98E+02	4.10E+02	4.45E+02	2.13E+01
C-17-f26	2.00E+02	3.00E+02	2.94E+02	3.00E+02	2.37E+01
C-17-f27	3.89E+02	3.90E+02	3.95E+02	4.05E+02	3.40E+00
C-17-f28	0.00E+00	3.00E+02	3.97E+02	6.11E+02	1.49E+02
C-17-f29	2.30E+02	2.49E+02	2.52E+02	2.85E+02	1.38E+01
C-17-f30	3.97E+02	4.57E+02	4.58E+02	6.74E+02	1.10E+02

Table 7
Statistical results of WSO over 51 independent runs on the CEC-2017 functions of 30 variables with 300,000 FEs.

F	Best	Median	Ave	Worst	Std
C-17-f1	2.89E−06	2.04E−04	1.37E−03	3.06E−02	4.39E−03
C-17-f3	4.24E−05	1.82E−03	7.65E−03	1.20E−01	1.92E−02
C-17-f4	0.00E+00	3.98E+00	1.32E+00	1.35E+01	1.15E+01
C-17-f5	3.28E+01	6.26E+01	6.35E+01	1.18E+02	1.80E+01
C-17-f6	2.41E−05	5.94E−01	1.44E+00	9.62E+00	1.93E+00
C-17-f7	7.75E+01	1.37E+02	1.45E+02	2.66E+02	4.26E+01
C-17-f8	3.58E+01	6.66E+01	6.54E+01	9.75E+01	1.67E+01
C-17-f9	1.22E+02	3.58E+02	2.91E+02	1.68E+03	3.54E+02
C-17-f10	1.83E+03	2.86E+03	2.94E+03	4.49E+03	2.21E+02
C-17-f11	4.27E+01	1.06E+02	1.04E+02	2.09E+02	4.23E+01
C-17-f12	2.55E+03	1.40E+04	1.92E+04	5.36E+04	1.35E+04
C-17-f13	3.94E+01	1.51E+02	1.96E+02	9.32E+02	1.53E+02
C-17-f14	7.17E+01	1.24E+02	1.27E+02	2.17E+02	3.56E+01
C-17-f15	3.49E+01	1.39E+02	1.49E+02	2.95E+02	5.10E+01
C-17-f16	1.73E+01	5.77E+02	5.53E+02	1.26E+03	2.29E+02
C-17-f17	6.58E+01	1.85E+02	1.80E+02	4.87E+02	8.08E+01
C-17-f18	4.52E+01	1.49E+02	3.45E+02	1.08E+03	7.28E+02
C-17-f19	2.22E+01	7.48E+01	7.67E+01	1.59E+02	3.08E+01
C-17-f20	8.68E+01	1.88E+02	1.80E+02	4.12E+02	7.37E+01
C-17-f21	2.42E+02	2.73E+02	2.73E+02	3.33E+02	1.61E+01
C-17-f22	1.00E+02	1.00E+02	1.02E+02	1.19E+02	2.86E+00
C-17-f23	2.26E+02	3.01E+02	3.12E+02	4.68E+02	3.22E+01
C-17-f24	4.16E+02	4.96E+02	5.02E+02	6.77E+02	4.53E+01
C-17-f25	3.83E+02	3.92E+02	3.86E+02	4.54E+02	2.03E+01
C-17-f26	2.00E+02	2.06E+03	1.52E+03	4.00E+03	1.16E+03
C-17-f27	5.23E+02	5.72E+02	5.74E+02	6.54E+02	2.97E+01
C-17-f28	3.00E+02	3.00E+02	3.23E+02	4.13E+02	4.45E+01
C-17-f29	3.91E+02	5.60E+02	5.66E+02	9.12E+02	1.01E+02
C-17-f30	1.10E+03	2.38E+03	2.95E+03	1.01E+04	1.56E+03

above corroborate that the proposed behavior model of WSO is consistently capable of exploring and exploiting the neighboring areas of both motionless and moving prey.

4. Experimental results and comparisons

This section presents the computational results of WSO on publicly available benchmark optimization problems. The results of WSO are discussed, analyzed and compared with other state-of-the-art meta-heuristics that have reported encouraging performance in the literature.

4.1. Description of the benchmark functions

The performance of the proposed WSO was tested on 29 benchmark test functions used in the CEC-2017 special sessions on real-parameter optimization. The CEC-2017 benchmark test set consists of 29 stable test functions and one unstable test function. These test functions form very challenging test beds that include both hybrid and composite test functions, in which they were created through rotation, shifting, expansion and hybridization of multimodal and unimodal functions. These functions simulate the complexity of a real search space that contains a big number of local optimums with diverse shapes of functions in different areas. These test cases were designed to assess local optimal avoidance accuracy as well as to examine the exploration

Table 8
Statistical results of WSO over 51 independent runs on the CEC-2017 functions of 50 variables with 500,000 FEs.

F	Best	Median	Ave	Worst	Std
C-17-f1	6.39E−02	5.62E+00	2.29E+01	1.06E+02	3.73E+01
C-17-f3	1.55E+01	1.65E+02	1.97E+02	7.66E+02	2.18E+01
C-17-f4	3.48E−03	5.85E+01	6.59E+01	1.17E+02	4.30E+01
C-17-f5	1.66E+02	1.88E+02	2.05E+02	2.68E+02	3.65E+01
C-17-f6	1.63E+00	3.68E+00	4.80E+00	1.05E+01	3.02E+00
C-17-f7	2.68E+02	4.40E+02	4.23E+02	5.39E+02	8.17E+01
C-17-f8	1.69E+02	1.97E+02	2.00E+02	2.47E+02	2.57E+01
C-17-f9	3.00E+03	5.15E+03	4.88E+03	6.37E+03	9.58E+02
C-17-f10	4.06E+03	5.14E+03	5.19E+03	6.21E+03	7.09E+02
C-17-f11	1.91E+02	2.37E+02	2.40E+02	2.99E+02	3.53E+01
C-17-f12	3.51E+04	9.82E+04	9.78E+04	1.66E+05	4.47E+04
C-17-f13	1.00E+03	3.84E+03	3.36E+03	1.21E+04	2.24E+03
C-17-f14	1.04E+02	1.22E+02	1.17E+02	1.99E+02	5.54E+01
C-17-f15	1.85E+02	2.90E+02	2.95E+02	1.07E+03	2.29E+02
C-17-f16	5.87E+02	1.14E+03	1.33E+03	2.24E+03	5.51E+02
C-17-f17	7.35E+02	1.05E+03	1.09E+03	1.44E+03	2.45E+02
C-17-f18	1.05E+02	1.36E+03	1.25E+03	1.85E+03	5.74E+02
C-17-f19	1.08E+02	1.26E+03	1.54E+03	3.60E+03	1.25E+03
C-17-f20	2.39E+02	5.18E+02	5.10E+02	7.84E+02	1.85E+02
C-17-f21	4.05E+02	5.09E+02	5.06E+02	6.37E+02	7.60E+01
C-17-f22	7.12E+02	3.08E+03	2.93E+03	6.57E+03	1.05E+03
C-17-f23	1.07E+02	1.27E+02	2.16E+02	9.49E+02	1.19E+02
C-17-f24	1.18E+03	1.52E+03	1.45E+03	2.71E+03	1.36E+02
C-17-f25	4.21E+02	5.53E+02	5.50E+02	6.02E+02	2.32E+01
C-17-f26	2.00E+02	1.44E+03	1.39E+03	6.92E+03	1.98E+03
C-17-f27	1.21E+02	4.66E+02	4.82E+02	9.23E+02	1.11E+02
C-17-f28	4.54E+02	4.78E+02	4.85E+02	6.60E+02	3.29E+01
C-17-f29	8.71E+02	1.79E+03	1.54E+03	3.14E+03	2.15E+02
C-17-f30	1.82E+05	6.31E+05	6.29E+05	1.11E+06	1.14E+05

Table 9
Statistical results of WSO over 51 independent runs on the CEC-2017 functions of 100 variables with 100,0000 FEs.

F	Best	Median	Ave	Worst	Std
C-17-f1	1.01E+03	7.77E+03	7.72E+03	9.82E+03	2.13E+03
C-17-f3	9.35E+02	2.59E+03	2.84E+03	5.63E+03	1.36E+03
C-17-f4	1.31E+02	2.26E+02	2.16E+02	2.95E+02	5.50E+01
C-17-f5	5.07E+02	5.88E+02	5.76E+02	8.40E+02	6.00E+01
C-17-f6	8.49E+00	1.37E+01	1.33E+01	4.23E+01	4.56E+00
C-17-f7	1.71E+02	2.92E+02	2.88E+02	4.98E+02	1.12E+02
C-17-f8	4.69E+02	6.52E+02	6.63E+02	8.02E+02	1.28E+02
C-17-f9	1.42E+04	1.60E+04	1.57E+04	1.98E+04	1.33E+03
C-17-f10	9.98E+03	1.03E+04	1.10E+04	1.46E+04	1.46E+03
C-17-f11	4.78E+02	5.83E+02	5.40E+02	8.61E+02	1.26E+02
C-17-f12	3.68E+05	8.52E+05	7.11E+05	8.13E+05	2.98E+05
C-17-f13	3.26E+03	4.77E+03	5.88E+03	7.60E+03	3.30E+03
C-17-f14	9.27E+03	1.67E+04	1.48E+04	1.86E+04	4.93E+03
C-17-f15	1.00E+03	2.08E+03	2.76E+03	5.21E+03	2.18E+03
C-17-f16	9.74E+02	3.33E+03	3.41E+03	4.36E+03	6.04E+02
C-17-f17	2.74E+03	3.42E+03	3.34E+03	4.16E+03	4.29E+02
C-17-f18	2.12E+04	3.70E+04	3.85E+04	5.13E+04	1.42E+04
C-17-f19	4.23E+02	9.77E+02	1.58E+03	4.26E+03	1.63E+03
C-17-f20	1.55E+03	2.32E+03	2.39E+03	3.46E+03	5.65E+02
C-17-f21	1.16E+03	1.21E+03	1.25E+03	1.49E+03	1.01E+02
C-17-f22	1.33E+04	1.50E+04	1.57E+04	2.28E+04	2.45E+03
C-17-f23	4.29E+02	7.50E+02	7.51E+02	8.26E+02	2.12E+02
C-17-f24	2.00E+03	2.44E+03	2.39E+03	9.83E+03	3.16E+02
C-17-f25	6.96E+02	8.25E+02	7.90E+02	9.35E+02	5.87E+01
C-17-f26	5.89E+02	8.29E+03	8.18E+03	1.09E+04	1.05E+03
C-17-f27	6.35E+02	8.90E+02	8.87E+02	1.11E+03	1.44E+02
C-17-f28	4.68E+02	5.53E+02	5.64E+02	8.98E+02	8.74E+01
C-17-f29	3.20E+03	4.36E+03	4.20E+03	5.51E+03	6.762E+02
C-17-f30	3.59E+03	1.14E+04	1.19E+04	2.68E+04	7.74E+03

capacity of optimization algorithms. It is widely known that a proficient optimization method should evade local optimal solutions and swiftly converge to the global optimum. Therefore, to study the robustness of WSO, this test set was used as it presents difficult problems and is used to add a significant challenge when evaluating the performance of WSO. Further details about the CEC-2017 benchmark functions can be found below and in [106]. To add further challenge to the performance of WSO on real-world optimization problems, it was evaluated on the CEC-2011 test suite. This set consists of 22 test problems for the CEC2011 special session and competition on real-world optimization problems. A detailed description of these test problems can be found in [107].

4.2. Experimental setup and involved algorithms

To bear out the general performance and comprehensive evaluation of the proposed WSO, its results are compared with nine highly respected optimization algorithms in the relevant literature when tested on the test suites mentioned above. The competitor algorithms to the proposed one fall into three main classes, namely: (i) GA [108], DE [22], BBO [109] and SFS [27] as the most studied EAs, (ii) PSO [110], ACO [31] and AMO [51] as hot and reliable SI algorithms, and (iii) TLBO [69] and GSK [1] as efficacious and recent human-based optimizers. The control parameters and settings of WSO and other competing algorithms are shown in Table 5.

The parameter settings of the rival algorithms reported in Table 5 are as given in [1], which were taken directly from their native references as mentioned in [1]. The initialization process of WSO is analogous to that used by other competing algorithms. This is for an equitable comparison between WSO and those competitor algorithms. The average (Ave) and standard deviation (Std) score metrics were used as performance assessment indicators for the accuracy and stability of the algorithms. In this work, these statistical evaluation measures were computed for each algorithm in each function as the two best solutions. The mean measure was used to evaluate the accuracy of the algorithms, while the analysis of the standard deviation results aims to ensure stable performance of the algorithms during the independent runs. The best results are reinforced for all test problems throughout the paper to give them more prominence out of other results. The following subsections present and discuss the performance of WSO compared to other optimization algorithms in CEC-2017 and CEC-2011 benchmark functions.

4.3. Performance of WSO on IEEE CEC-2017

With the goal of presenting a challenge to the proposed WSO, a recent benchmark test suite, CEC-2017, was employed. This test consists of 30 test functions involving rotated and shifted unimodal, multimodal, hybrid and composition functions [106,111]. These functions can be divided into four classes: (1) unimodal functions, namely C-17-f1–C-17-f3, (2) simple multimodal functions, namely C-17-f4–C-17-f10, (3) hybrid functions, namely C-17-f11–C-17-f20, and (4) composition functions, which are C-17-f21–C-17-f30. In view of this, the majority of these test functions are among the most difficult composition and hybrid functions. It should also be noted that due to the unstable behavior of C-17-f2 function particularly for higher dimensions, it was set aside from this set. The search range for all of these test functions is [−100, 100] considering the different dimensions of each test function as presented below.

The performance of WSO was evaluated on this test suite, where we adopted the solution error measure ($f(x)$ - $f(x*)$), where x is the best solution that the algorithm got in one run and $x*$ is the well-known global optimum for each test function [1]. For all of the test functions in the CEC-2017, WSO used 100 white sharks along with a maximum number of FEs of $10000 \times d$, bearing in mind that d represents the dimensions of the problems. The stop criterion for the proposed WSO was set to the maximum number of FEs, where it ran 51 times independently

Table 10
Optimization results of WSO and other algorithms over 51 independent runs on the CEC-2017 test functions of 10 variables with 100,000 FEs.

F		WSO	TLBO	SFS	DE	GA	GSK	AMO	PSO	BBO	ACO
C-17-f1	Ave	**0.00E+00**	1.96E+03	5.60E+03	**0.00E+00**	1.14E+06	**0.00E+00**	9.84E+00	2.30E+03	1.12E+06	1.57E+10
	Std	**0.00E+00**	2.52E+03	3.26E+03	**0.00E+00**	2.18E+05	**0.00E+00**	2.69E+01	3.05E+03	2.26E+05	3.75E+09
C-17-f3	Ave	**0.00E+00**	**0.00E+00**	2.05E−02	**0.00E+00**	1.24E+04	**0.00E+00**	2.90E−08	**0.00E+00**	8.35E+02	6.32E+03
	Std	**0.00E+00**	**0.00E+00**	1.17E−02	**0.00E+00**	7.87E+03	**0.00E+00**	9.16E−08	**0.00E+00**	8.07E+02	1.74E+03
C-17-f4	Ave	**0.00E+00**	1.79E−01	9.37E−01	**0.00E+00**	1.17E+01	**0.00E+00**	1.39E+00	2.82E+00	6.20E+00	1.83E+02
	Std	**0.00E+00**	3.45E−01	6.74E−01	**0.00E+00**	3.29E+00	**0.00E+00**	6.29E−01	1.21E+00	1.72E+00	4.26E+01
C-17-f5	Ave	8.52E+00	8.24E+00	8.68E+00	2.48E+01	4.90E+01	2.03E+01	**6.36E+00**	1.59E+01	7.96E+00	6.86E+01
	Std	3.21E+00	3.38E+00	3.30E+00	4.13E+00	1.22E+01	2.79E+00	**1.49E+00**	7.08E+00	2.77E+00	8.53E+00
C-17-f6	Ave	1.40E−07	3.76E−02	5.35E−03	**0.00E+00**	1.24E+01	**0.00E+00**	**0.00E+00**	8.27E−02	9.19E−01	3.57E+01
	Std	5.67E−07	1.28E−01	1.59E−03	**0.00E+00**	4.54E+00	**0.00E+00**	**0.00E+00**	3.36E−01	**0.00E+00**	5.40E+00
C-17-f7	Ave	1.79E+01	1.76E+01	2.33E+01	3.43E+01	6.87E+01	3.07E+01	1.82E+01	**1.72E+01**	2.31E+01	1.23E+02
	Std	4.60E+00	3.25E+00	3.60E+00	4.84E+00	1.46E+01	3.08E+00	**1.42E+00**	4.46E+00	3.82E+00	1.43E+01
C-17-f8	Ave	8.72E+00	**6.84E+00**	7.42E+00	2.37E+01	3.54E+01	2.02E+01	**6.84E+00**	1.23E+01	8.66E+00	4.85E+01
	Std	4.92E+00	2.75E+00	2.76E+00	3.76E+00	8.94E+00	2.92E+00	**1.49E+00**	5.33E+00	3.05E+00	6.66E+00
C-17-f9	Ave	1.76E−02	2.64E−01	6.68E−06	**0.00E+00**	2.39E+01	**0.00E+00**	**0.00E+00**	**0.00E+00**	3.15E−01	5.56E+02
	Std	7.02E−02	4.22E−01	4.34E−06	**0.00E+00**	1.75E+01	**0.00E+00**	**0.00E+00**	**0.00E+00**	8.16E−02	1.53E+02
C-17-f10	Ave	5.44E+02	4.95E+02	3.63E+02	4.94E+02	7.52E+02	1.06E+03	3.65E+02	6.30E+02	**2.57E+02**	1.45E+03
	Std	2.84E+02	2.92E+02	1.91E+02	2.99E+02	2.36E+02	1.33E+02	**1.01E+02**	2.64E+02	1.55E+02	1.34E+02
C-17-f11	Ave	4.19E+00	7.06E+00	4.61E+00	4.33E−02	8.21E+01	**0.00E+00**	2.76E+00	1.10E+01	8.92E+00	1.96E+02
	Std	3.49E+00	5.17E+00	1.25E+00	1.95E−01	7.89E+01	**0.00E+00**	7.90E−01	7.27E+00	4.26E+00	6.99E+01
C-17-f12	Ave	3.66E+02	1.28E+04	5.04E+03	**8.06E+00**	1.12E+06	8.93E+01	1.13E+04	1.33E+04	2.55E+05	9.79E+08
	Std	1.77E+02	9.07E+03	2.11E+03	**1.89E+01**	1.78E+06	7.26E+01	6.49E+03	1.24E+04	1.60E+05	9.01E+08
C-17-f13	Ave	2.34E+01	1.70E+03	4.55E+01	**6.56E+00**	2.62E+04	**6.56E+00**	2.67E+01	6.45E+03	8.09E+04	7.02E+06
	Std	1.93E+01	1.91E+03	9.83E+00	1.86E+00	2.19E+04	**1.41E+00**	8.99E+00	5.72E+03	7.12E+04	2.36E+07
C-17-f14	Ave	2.15E+01	3.40E+01	2.20E+01	**4.30E−02**	1.48E+04	5.88E+00	3.83E+00	4.57E+01	1.20E+04	4.33E+02
	Std	1.28E+01	7.43E+00	3.82E+00	**1.96E−01**	6.43E+03	3.06E+00	1.67E+00	1.93E+01	1.01E+04	1.10E+02
C-17-f15	Ave	6.57E+00	4.75E+01	1.00E+01	**3.60E−02**	1.38E+04	2.22E−01	1.75E+00	5.62E+01	1.95E+04	3.74E+03
	Std	5.39E+00	2.09E+01	2.14E+00	**1.03E−01**	1.26E+04	2.13E−01	5.40E−01	5.89E+01	1.94E+04	2.37E+03
C-17-f16	Ave	1.27E+01	1.44E+01	4.17E+00	2.93E+00	3.00E+01	4.27E+00	**1.13E+00**	2.05E+02	1.06E+02	2.65E+02
	Std	3.57E+01	3.48E+01	3.16E+00	4.65E+00	4.18E+01	4.95E+00	**2.64E−01**	1.19E+02	8.75E+01	5.71E+01
C-17-f17	Ave	2.52E+01	2.76E+01	2.34E+01	**4.82E+00**	4.12E+01	1.17E+01	4.92E+00	4.56E+01	2.74E+01	1.14E+02
	Std	1.25E+01	8.83E+00	5.66E+00	6.77E+00	1.29E+01	7.09E+00	**2.53E+00**	2.30E+01	3.49E+01	2.31E+01
C-17-f18	Ave	2.15E+01	4.55E+03	5.22E+01	**8.18E−02**	8.30E+04	3.20E−01	3.13E+01	5.10E+03	1.45E+05	6.67E+07
	Std	1.12E+01	3.67E+03	1.05E+01	**1.73E−01**	9.59E+04	1.92E−01	8.02E+00	5.76E+03	1.01E+05	1.65E+08
C-17-f19	Ave	2.73E+00	2.94E+01	5.84E+00	**6.27E−03**	6.30E+04	1.55E−01	1.11E+00	9.57E+01	4.34E+04	1.24E+04
	Std	2.24E+00	1.83E+01	8.94E−01	**1.11E−02**	4.21E+04	3.51E−01	4.28E−01	2.95E+02	4.02E+04	1.31E+04
C-17-f20	Ave	2.43E+01	1.75E+01	1.21E+01	1.10E−01	1.02E+02	1.19E+00	**6.19E−04**	5.51E+01	1.29E+01	1.20E+02
	Std	1.13E+01	1.04E+01	3.31E+00	1.63E−01	5.01E+01	3.99E+00	**1.74E−03**	5.42E+01	5.91E+00	2.77E+01
C-17-f21	Ave	1.68E+02	1.62E+02	**1.00E+02**	1.80E+02	1.54E+02	1.93E+02	1.37E+02	1.79E+02	2.00E+02	1.37E+02
	Std	5.31E+01	5.26E+01	**5.06E−02**	6.22E+01	1.49E+01	5.07E+01	5.02E+01	5.65E+01	3.07E+01	1.29E+01
C-17-f22	Ave	9.40E+01	9.98E+01	**9.24E+01**	9.48E+01	1.25E+02	1.00E+02	9.91E+01	9.38E+01	1.06E+02	2.57E+02
	Std	2.47E+01	1.07E+01	3.00E+01	2.30E+01	2.17E+01	**4.87E−01**	7.18E+00	2.55E+01	7.47E+00	6.14E+01
C-17-f23	Ave	3.11E+02	3.10E+02	**3.03E+02**	3.17E+02	3.63E+02	3.18E+02	3.08E+02	3.28E+02	3.13E+02	3.78E+02
	Std	5.34E+00	4.31E+00	4.35E+01	5.20E+00	1.51E+01	3.99E+00	**1.61E+00**	1.24E+01	4.97E+00	9.61E+00
C-17-f24	Ave	2.88E+02	3.23E+02	**2.18E+02**	3.43E+02	3.90E+02	3.44E+02	2.85E+02	3.24E+02	3.36E+02	2.93E+02
	Std	9.43E+01	5.62E+01	1.18E+02	4.97E+01	3.40E+01	**1.87E+01**	8.09E+01	8.33E+01	2.66E+01	3.63E+01
C-17-f25	Ave	**4.10E+02**	4.26E+02	4.21E+02	4.11E+02	4.30E+02	4.27E+02	4.18E+02	4.25E+02	4.34E+02	5.76E+02
	Std	2.13E+01	2.23E+01	2.30E+01	2.10E+01	2.52E+01	**2.05E+01**	2.23E+01	2.29E+01	2.26E+01	4.23E+01
C-17-f26	Ave	2.94E+02	3.31E+02	2.92E+02	3.00E+02	4.19E+02	3.00E+02	3.00E+02	**2.74E+02**	3.40E+02	7.47E+02
	Std	2.37E+01	5.60E+01	4.40E+01	**0.00E+00**	8.88E+01	**0.00E+00**	**0.00E+00**	7.63E+01	1.51E+02	8.18E+01
C-17-f27	Ave	3.95E+02	3.93E+02	3.92E+02	3.90E+02	4.10E+02	**3.89E+02**	3.91E+02	4.03E+02	3.97E+02	4.40E+02
	Std	3.40E+00	2.62E+00	1.85E+00	2.63E−01	9.10E+00	**2.17E−01**	2.38E+00	1.97E+01	4.39E+00	8.57E+00
C-17-f28	Ave	3.97E+02	4.01E+02	3.06E+02	3.63E+02	5.82E+02	3.12E+02	**2.99E+02**	4.54E+02	5.48E+02	5.85E+02
	Std	1.49E+02	1.26E+02	3.97E+01	1.21E+02	1.56E+02	3.20E+01	**3.99E+00**	1.57E+02	9.68E+01	4.64E+01
C-17-f29	Ave	2.52E+02	2.63E+02	2.59E+02	**2.40E+02**	3.04E+02	2.48E+02	2.64E+02	3.05E+02	2.65E+02	3.90E+02
	Std	1.38E+01	1.46E+01	1.18E+01	**4.73E+00**	4.09E+01	4.76E+00	8.17E+00	4.50E+01	1.56E+01	3.41E+01
C-17-f30	Ave	4.58E+02	1.25E+05	2.03E+03	1.64E+04	3.73E+05	**4.56E+02**	7.42E+03	2.00E+05	4.63E+05	2.23E+07
	Std	1.10E+02	2.98E+05	1.63E+03	1.14E+05	3.78E+05	**3.44E+01**	5.17E+03	3.79E+05	5.39E+05	2.32E+07

for each function in each experiment. The parameter settings of the proposed WSO are given in Table 5. For this test, the dimensions of the test functions considered in this work are 10, 30, 50 and 100. The statistical results of WSO on the CEC-2017 benchmark test functions with dimensions of 10, 30, 50 and 100 are shown in Tables 6, 7, 8 and 9, respectively. These results are presented in terms of the best, median, average, worst and standard deviation values of error from optimal solution over a number of 51 independent runs for all 29 test functions.

As it is presented in Tables 6, 7, 8 and 9, WSO has reported sensible solution rates in solving complex optimization problems with different dimensions. Overall, it can be obviously observed that WSO has excellent performance on unimodal problems (C-17-f1 and C-17-f3) and is capable of finding the global optimum solution continuously over 51 independent runs in 10 dimensions. However, in problems with dimensions 30, 50 and 100, the optimal was not detected on unimodal problems in any case as observed in Tables 7, 8 and 9, respectively. In any case, the mean error and standard deviation values are relatively small in 30d and relatively large in 50d and 100d, but still within acceptable ranges of optimization for such high dimensional problems. In the multimodal functions (C-17-f4-C-17-f10), the optimal solution is revealed in problem C-17-f4 in 10d, but it is not revealed in any problem in dimensions 30, 50 and 100. However, the mean error and standard deviation values, in these cases, are relatively small and not far from the global optimum solutions, especially for problems C-17-f6 and C-17-f9. In the hybrid functions (C-17-f11–C-17-f20), WSO is competent to get good solutions for all

Table 11
Optimization results of WSO and other algorithms over 51 independent runs on the CEC-2017 test functions of 30 variables with 300,000 FEs.

F		WSO	TLBO	SFS	DE	GA	GSK	AMO	PSO	BBO	ACO
C-17-f1	Ave	1.37E−03	3.36E+03	3.17E+03	**0.00E+00**	6.06E+06	**0.00E+00**	5.34E+00	3.43E+03	4.20E+06	8.95E+10
	Std	4.39E−03	3.19E+03	3.76E+03	**0.00E+00**	1.62E+06	**0.00E+00**	7.57E+00	3.97E+03	1.87E+05	2.59E+10
C-17-f3	Ave	7.65E−03	1.03E−04	5.36E+01	1.36E+02	6.81E+04	**7.07E−07**	4.56E+03	1.48E+02	4.39E+04	2.08E+07
	Std	1.92E−02	2.38E−04	2.72E+01	1.36E+02	2.02E+04	**1.82E−06**	1.48E+03	5.90E+01	2.58E+04	1.23E+08
C-17-f4	Ave	**1.32E+00**	5.63E+01	5.91E+01	5.92E+01	1.51E+02	1.11E+01	4.60E+00	8.59E+01	9.99E+01	9.61E+03
	Std	1.15E+01	3.51E+01	4.04E+01	**1.81E+00**	4.09E+01	2.17E+01	1.07E+01	3.22E+01	2.27E+01	1.72E+03
C-17-f5	Ave	6.35E+01	9.06E+01	6.76E+01	1.79E+02	2.26E+02	1.60E+02	5.41E+01	1.14E+02	**4.21E+01**	4.11E+02
	Std	1.80E+01	2.00E+01	1.34E+01	1.27E+01	2.93E+01	8.87E+00	**6.68E+00**	2.67E+01	1.06E+01	2.27E+01
C-17-f6	Ave	1.44E+00	8.45E+00	1.18E−02	**0.00E+00**	3.86E+01	1.52E−06	**0.00E+00**	3.09E+00	8.98E−01	8.31E+01
	Std	1.93E+00	4.18E+00	2.04E−02	**0.00E+00**	9.95E+00	2.36E−06	**0.00E+00**	4.01E+00	4.07E−02	6.11E+00
C-17-f7	Ave	1.45E+02	1.48E+02	1.01E+02	2.12E+02	3.24E+02	1.87E+02	**9.15E+01**	9.71E+01	1.14E+02	9.44E+02
	Std	4.26E+01	3.18E+01	2.19E+01	9.99E+00	5.43E+01	8.40E+00	**6.90E+00**	1.66E+01	1.46E+01	9.83E+01
C-17-f8	Ave	6.54E+01	7.06E+01	7.50E+01	1.80E+02	2.43E+02	1.55E+02	5.42E+01	9.73E+01	**4.31E+01**	3.63E+02
	Std	1.67E+01	1.44E+01	1.96E+01	1.14E+01	3.07E+01	1.11E+01	**5.76E+00**	2.11E+01	1.00E+01	1.97E+01
C-17-f9	Ave	2.91E+02	2.07E+02	2.47E+01	**0.00E+00**	1.12E+03	**0.00E+00**	1.05E−01	1.01E+03	1.09E+02	1.39E+04
	Std	3.54E+02	1.29E+02	6.83E+01	**0.00E+00**	1.42E+03	**0.00E+00**	2.07E−01	1.15E+03	8.42E+01	1.46E+03
C-17-f10	Ave	2.94E+03	6.01E+03	2.35E+03	6.61E+03	4.48E+03	6.69E+03	3.62E+03	3.20E+03	**2.29E+03**	7.38E+03
	Std	**2.21E+02**	1.17E+03	5.20E+02	4.36E+02	8.44E+02	3.54E+02	2.66E+02	5.52E+02	4.48E+02	2.24E+02
C-17-f11	Ave	1.04E+02	1.36E+02	4.57E+01	7.49E+01	1.33E+03	**3.30E+01**	4.91E+01	1.14E+02	1.43E+03	5.54E+03
	Std	4.23E+01	4.75E+01	2.85E+01	3.10E+01	1.01E+03	3.83E+01	**2.47E+01**	3.55E+01	1.40E+03	1.53E+03
C-17-f12	Ave	1.92E+04	4.44E+04	2.75E+04	7.55E+03	4.29E+06	**6.62E+03**	6.07E+04	1.43E+05	3.52E+06	1.89E+10
	Std	1.35E+04	6.48E+04	1.38E+04	7.17E+03	3.34E+06	**4.59E+03**	6.88E+04	9.98E+04	2.19E+06	4.59E+09
C-17-f13	Ave	1.96E+02	1.40E+04	9.39E+02	**8.24E+01**	2.54E+06	9.83E+01	6.78E+03	1.49E+04	1.51E+06	1.23E+10
	Std	1.53E+02	1.41E+04	2.79E+02	**9.84E+00**	2.22E+06	3.42E+01	3.23E+03	1.63E+04	4.77E+05	7.17E+09
C-17-f14	Ave	1.27E+02	2.99E+03	8.72E+01	6.33E+01	9.85E+05	**5.69E+01**	2.22E+03	1.05E+04	8.23E+05	9.44E+06
	Std	3.56E+01	2.58E+03	9.94E+00	**4.72E+00**	9.57E+05	5.49E+00	1.39E+03	8.32E+03	8.54E+05	2.25E+07
C-17-f15	Ave	1.49E+02	5.17E+03	1.52E+02	3.91E+01	7.21E+05	**1.45E+01**	4.22E+02	7.96E+03	7.01E+05	3.56E+09
	Std	5.10E+01	6.89E+03	2.55E+01	**5.79E+00**	2.53E+05	1.49E+01	5.61E+02	8.62E+03	3.26E+05	2.14E+09
C-17-f16	Ave	5.53E+02	5.67E+02	**4.41E+02**	7.78E+02	1.17E+03	7.96E+02	5.52E+02	8.59E+02	8.95E+02	2.90E+03
	Std	2.29E+02	2.18E+02	1.58E+02	4.11E+02	3.00E+02	1.94E+02	**1.19E+02**	2.38E+02	3.11E+02	2.10E+02
C-17-f17	Ave	1.80E+02	2.00E+02	1.05E+02	1.02E+02	6.43E+02	1.89E+02	**8.49E+01**	3.57E+02	3.81E+02	1.28E+03
	Std	8.08E+01	9.15E+01	6.29E+01	5.03E+01	2.04E+02	9.44E+01	**1.77E+01**	1.64E+02	2.16E+02	1.66E+02
C-17-f18	Ave	3.45E+02	2.05E+05	3.32E+02	3.83E+01	3.77E+06	**3.68E+01**	1.43E+05	1.84E+05	1.95E+06	1.78E+08
	Std	7.28E+02	1.40E+05	7.40E+01	**4.10E+00**	4.84E+06	5.42E+00	5.21E+04	1.36E+05	1.96E+06	1.13E+08
C-17-f19	Ave	7.67E+01	5.57E+03	5.83E+01	1.89E+01	8.32E+05	**1.29E+01**	1.32E+03	7.77E+03	8.37E+05	3.91E+09
	Std	3.08E+01	6.13E+03	9.10E+00	**5.75E+00**	3.31E+05	6.02E+00	1.53E+03	1.12E+04	3.40E+05	2.60E+09
C-17-f20	Ave	1.80E+02	2.20E+02	1.30E+02	**6.03E+01**	4.10E+02	1.08E+02	1.42E+02	3.78E+02	4.33E+02	8.37E+02
	Std	7.37E+01	7.33E+01	6.05E+01	6.50E+01	1.11E+02	1.14E+02	**4.76E+01**	1.36E+02	1.85E+02	1.02E+02
C-17-f21	Ave	2.73E+02	2.70E+02	2.56E+02	3.66E+02	4.57E+02	3.46E+02	2.53E+02	3.06E+02	**2.49E+02**	5.90E+02
	Std	1.61E+01	1.83E+01	2.71E+01	1.34E+01	3.28E+01	8.30E+00	**6.95E+00**	2.35E+01	1.04E+01	1.72E+01
C-17-f22	Ave	1.02E+02	2.08E+02	**1.00E+02**	1.81E+03	5.16E+03	**1.00E+02**	**1.00E+02**	9.62E+02	1.58E+03	5.74E+03
	Std	2.86E+00	7.55E+02	5.50E−04	2.95E+03	1.60E+03	**0.00E+00**	**0.00E+00**	1.61E+03	1.49E+03	4.29E+02
C-17-f23	Ave	**3.12E+02**	4.42E+02	4.08E+02	5.26E+02	6.77E+02	4.70E+02	3.98E+02	5.08E+02	4.02E+02	9.40E+02
	Std	3.22E+01	2.62E+01	1.35E+01	9.51E+00	3.70E+01	4.49E+01	**9.02E+00**	5.60E+01	1.24E+01	4.40E+01
C-17-f24	Ave	5.02E+02	4.97E+02	4.93E+02	5.96E+02	8.14E+02	5.68E+02	**4.64E+02**	5.73E+02	4.67E+02	1.06E+03
	Std	4.53E+01	2.37E+01	2.27E+01	**7.08E+00**	6.40E+01	1.45E+01	8.44E+00	6.32E+01	1.33E+01	5.58E+01
C-17-f25	Ave	**3.86E+02**	4.08E+02	**3.86E+02**	3.87E+02	5.61E+02	3.87E+02	3.87E+02	3.90E+02	3.93E+02	3.49E+03
	Std	2.03E+01	2.28E+01	2.68E+00	**2.19E−02**	9.50E+01	2.11E−01	1.07E+00	8.76E+00	9.81E+00	6.51E+02
C-17-f26	Ave	1.52E+03	2.10E+03	1.32E+03	2.58E+03	3.61E+03	9.87E+02	1.49E+03	**1.27E+03**	1.63E+03	7.10E+03
	Std	1.16E+03	1.08E+03	8.12E+02	2.61E+02	8.15E+02	2.49E+02	1.95E+02	1.39E+03	**1.56E+02**	4.27E+02
C-17-f27	Ave	5.74E+02	5.34E+02	5.16E+02	4.96E+02	6.13E+02	**4.93E+02**	5.16E+02	5.43E+02	5.26E+02	1.14E+03
	Std	2.97E+01	2.02E+01	1.25E+01	7.01E+00	4.04E+01	8.02E+00	**4.81E+00**	3.02E+01	7.66E+00	7.37E+01
C-17-f28	Ave	3.23E+02	3.85E+02	3.61E+02	3.20E+02	5.66E+02	3.21E+02	**3.14E+02**	4.14E+02	4.33E+02	3.51E+03
	Std	4.45E+01	5.63E+01	4.99E+01	4.41E+01	4.52E+01	4.22E+01	3.61E+01	2.58E+01	**2.47E+01**	4.77E+02
C-17-f29	Ave	5.66E+02	8.37E+02	5.53E+02	5.42E+02	9.07E+02	5.77E+02	**5.33E+02**	7.56E+02	7.18E+02	2.59E+03
	Std	1.01E+02	1.80E+02	1.04E+02	9.62E+01	1.78E+02	1.01E+02	**2.46E+01**	1.79E+02	1.45E+02	2.20E+02
C-17-f30	Ave	2.95E+03	5.24E+03	8.22E+03	**2.00E+03**	3.45E+05	2.08E+03	4.71E+03	5.95E+03	3.31E+05	3.16E+09
	Std	1.56E+03	2.76E+03	3.40E+03	**5.62E+01**	1.43E+05	9.27E+01	7.99E+02	2.82E+03	1.23E+05	9.12E+08

test cases, except for problems C-17-f10 and C-17f12 where the performance level was reduced when the dimension is increased. Lastly, regarding the composition functions (C-17-f21–C-17-f30), which are the most hard problems in the CEC-2017 benchmark test suite because they have different characteristics around a large number of local optima, the optimum is not revealed in any problem. However, the reported ranges of average errors and standard deviations are sensible for such highly complex, multimodal and non-separable benchmark test functions. As per the results in Tables 8 and 9, WSO is often fell in local optimum, but is still not far away from the global optimum in all test functions. Consequently, it can be deduced that in all test cases for all the dimensions considered, the difference margins between the mean and median measures are small even in cases where the final outcomes are far away from the optimal. Besides, WSO obtained small standard deviation results in the majority of test functions, which indicates that the robustness of WSO is stable. In short, the results in Tables 6, 7, 8 and 9, point out that WSO is a powerful meta-heuristic optimization algorithm. Lastly, due to the small difference between the results in the four dimensions considered, it can be said that the performance level of WSO decreases moderately, and it is still more robust and stable against the curse of dimensionality, that is, it is generally stable as the dimensions of the test problems grow.

For a more accurate judgment of the performance of WSO on this test suite, the collected results of WSO, in terms of mean errors and standard deviations, are compared with the algorithms considered above. For all of the test functions in the CEC-2017

Table 12
Optimization results of WSO and other algorithms over 51 independent runs on the CEC-2017 test functions of 50 variables with 500,000 FEs.

F		WSO	TLBO	SFS	DE	GA	GSK	AMO	PSO	BBO	ACO
C-17-f1	Ave	2.29E+01	2.56E+03	3.65E+03	**7.43E−01**	1.93E+07	1.09E+03	1.30E+03	3.87E+03	6.10E+06	1.94E+11
	Std	3.73E+01	3.07E+03	5.38E+03	**1.81E+00**	4.99E+06	1.24E+03	1.27E+03	6.33E+03	7.65E+05	5.32E+10
C-17-f3	Ave	**1.97E+02**	1.40E+03	4.44E+03	9.29E+04	8.00E+04	3.85E+03	3.15E+04	1.90E+03	1.23E+05	1.31E+06
	Std	**2.18E+01**	1.03E+03	1.28E+03	1.68E+04	1.78E+04	1.51E+03	4.71E+03	3.35E+02	4.22E+04	4.92E+06
C-17-f4	Ave	**6.59E+01**	1.03E+02	1.10E+02	8.03E+01	3.03E+02	8.33E+01	7.53E+01	1.57E+02	1.43E+02	3.28E+04
	Std	**4.30E+01**	4.48E+01	5.13E+01	4.94E+01	8.39E+01	5.00E+01	4.77E+01	4.67E+01	5.04E+01	3.36E+03
C-17-f5	Ave	2.05E+02	1.87E+02	2.13E+02	3.50E+02	4.47E+02	3.20E+02	1.34E+02	2.31E+02	**8.37E+01**	8.46E+02
	Std	3.65E+01	3.14E+01	4.68E+01	1.40E+01	5.07E+01	1.79E+01	**1.30E+01**	4.15E+01	1.39E+01	2.80E+01
C-17-f6	Ave	4.80E+00	2.17E+01	1.01E−01	**2.60E−07**	4.11E+01	3.78E−06	2.10E−04	1.55E+01	8.96E−01	1.05E+02
	Std	3.02E+00	5.12E+00	1.99E−01	**1.09E−06**	4.95E+00	3.52E−06	1.22E−03	1.18E+01	3.65E−02	4.46E+00
C-17-f7	Ave	4.23E+02	3.84E+02	2.56E+02	4.07E+02	5.87E+02	3.70E+02	2.00E+02	**1.99E+02**	2.30E+02	2.05E+03
	Std	8.17E+01	6.37E+01	5.48E+01	**1.09E+01**	6.93E+01	1.41E+01	1.64E+01	2.83E+01	2.38E+01	1.71E+02
C-17-f8	Ave	2.00E+02	2.02E+02	2.03E+02	3.53E+02	4.53E+02	3.24E+02	1.34E+02	2.33E+02	**8.71E+01**	8.06E+02
	Std	2.57E+01	2.96E+01	4.35E+01	1.42E+01	5.51E+01	1.36E+01	**1.35E+01**	3.73E+01	1.82E+01	2.61E+01
C-17-f9	Ave	4.88E+03	2.68E+03	2.90E+02	3.99E−02	3.96E+03	**1.07E−02**	3.94E+00	6.53E+03	5.69E+02	4.52E+04
	Std	9.58E+02	1.47E+03	4.52E+02	1.26E−01	2.17E+03	**2.79E−02**	4.75E+00	2.49E+03	3.49E+02	5.03E+03
C-17-f10	Ave	5.19E+03	1.02E+04	4.68E+03	1.30E+04	8.76E+03	1.30E+04	6.96E+03	5.56E+03	**4.11E+03**	1.36E+04
	Std	7.09E+02	2.67E+03	6.67E+02	7.47E+02	7.41E+02	4.50E+02	4.28E+02	8.76E+02	7.06E+02	**3.20E+02**
C-17-f11	Ave	2.40E+02	2.16E+02	1.28E+02	1.43E+02	6.84E+03	**3.45E+01**	9.62E+01	1.63E+02	3.99E+03	2.00E+04
	Std	3.53E+01	6.69E+01	3.60E+01	2.27E+01	4.28E+03	2.32E+01	**1.51E+01**	3.71E+01	3.26E+03	3.07E+03
C-17-f12	Ave	9.78E+04	5.90E+05	3.19E+05	6.19E+04	1.65E+07	**9.46E+03**	4.74E+05	1.87E+06	1.10E+07	1.11E+11
	Std	4.47E+04	1.31E+06	2.42E+05	3.75E+04	8.55E+06	**7.01E+03**	2.23E+05	1.32E+06	4.55E+06	1.89E+10
C-17-f13	Ave	3.36E+03	5.23E+03	3.50E+03	**5.33E+02**	2.17E+06	1.49E+03	1.45E+03	3.93E+03	2.12E+06	6.60E+10
	Std	2.24E+03	3.91E+03	3.20E+03	1.39E+03	5.56E+05	2.16E+03	**1.00E+03**	4.85E+03	7.10E+05	1.62E+10
C-17-f14	Ave	**1.17E+02**	4.92E+04	1.68E+02	1.25E+02	2.22E+06	1.24E+02	3.45E+04	5.32E+04	3.70E+06	8.63E+07
	Std	5.54E+01	4.29E+04	1.53E+01	**9.35E+00**	1.10E+06	1.87E+01	1.68E+04	3.68E+04	2.67E+06	3.34E+07
C-17-f15	Ave	2.95E+02	6.65E+03	3.00E+02	1.08E+02	1.59E+06	**4.20E+01**	2.24E+03	5.16E+03	1.49E+06	2.04E+10
	Std	2.29E+02	5.53E+03	4.76E+01	**1.00E+01**	3.13E+05	1.68E+01	1.63E+03	4.69E+03	4.43E+05	7.77E+09
C-17-f16	Ave	1.33E+03	1.17E+03	1.18E+03	2.51E+03	2.67E+03	1.83E+03	**1.05E+03**	1.50E+03	1.70E+03	5.80E+03
	Std	5.51E+02	3.09E+02	3.37E+02	6.16E+02	5.68E+02	6.59E+02	**1.69E+02**	3.84E+02	3.68E+02	3.21E+02
C-17-f17	Ave	1.09E+03	9.96E+02	7.49E+02	1.19E+03	1.26E+03	1.35E+03	**7.37E+02**	1.14E+03	1.20E+03	5.14E+03
	Std	2.45E+02	2.23E+02	1.96E+02	4.77E+02	3.48E+02	1.90E+02	**1.11E+02**	2.81E+02	3.13E+02	6.89E+02
C-17-f18	Ave	1.25E+03	5.45E+05	**4.53E+02**	7.67E+02	2.72E+06	5.98E+02	6.43E+05	1.02E+06	8.01E+06	5.13E+08
	Std	5.74E+02	3.28E+05	**1.55E+02**	1.31E+03	2.05E+06	3.37E+02	2.78E+05	5.61E+05	5.70E+06	5.72E+08
C-17-f19	Ave	1.54E+03	1.32E+04	1.22E+02	6.24E+01	5.69E+05	**3.05E+01**	1.01E+04	1.36E+04	6.51E+05	9.17E+09
	Std	1.25E+03	7.59E+03	2.92E+01	**6.16E+00**	2.18E+05	9.59E+00	3.71E+03	7.22E+03	1.96E+05	2.48E+09
C-17-f20	Ave	5.10E+02	5.81E+02	**5.01E+02**	9.98E+02	1.52E+03	1.37E+03	5.02E+02	9.06E+02	1.16E+03	2.05E+03
	Std	1.85E+02	2.76E+02	2.20E+02	5.60E+02	2.68E+02	1.28E+02	**9.84E+01**	2.68E+02	3.44E+02	1.01E+02
C-17-f21	Ave	5.06E+02	3.91E+02	3.47E+02	5.52E+02	7.20E+02	5.21E+02	3.27E+02	4.33E+02	**2.96E+02**	1.04E+03
	Std	7.60E+01	3.70E+01	3.36E+01	**1.28E+01**	6.23E+01	1.31E+01	1.58E+01	4.96E+01	1.58E+01	3.00E+01
C-17-f22	Ave	2.93E+03	7.62E+03	**2.44E+03**	1.32E+04	1.01E+04	1.10E+04	5.65E+03	6.02E+03	4.92E+03	1.38E+04
	Std	1.05E+03	5.62E+03	2.74E+03	**3.25E+02**	7.97E+02	4.85E+03	3.18E+03	2.58E+03	8.04E+02	3.66E+02
C-17-f23	Ave	**2.16E+02**	7.01E+02	5.98E+02	7.69E+02	9.94E+02	5.42E+02	5.60E+02	8.06E+02	5.48E+02	1.74E+03
	Std	1.19E+02	6.69E+01	4.60E+01	2.12E+01	4.44E+01	1.39E+02	**1.49E+01**	1.03E+02	2.41E+01	8.83E+01
C-17-f24	Ave	1.45E+03	7.25E+02	6.91E+02	8.48E+02	1.27E+03	6.34E+02	6.13E+02	8.83E+02	**5.88E+02**	1.96E+03
	Std	1.36E+02	5.00E+01	3.95E+01	**1.38E+01**	9.86E+01	1.39E+02	1.50E+01	8.98E+01	1.77E+01	1.08E+02
C-17-f25	Ave	5.50E+02	5.71E+02	5.60E+02	**4.96E+02**	6.90E+02	5.56E+02	5.62E+02	5.57E+02	5.75E+02	1.65E+04
	Std	**2.32E+01**	3.42E+01	2.79E+01	3.14E+01	6.08E+01	4.62E+01	2.87E+01	2.37E+01	2.68E+01	1.61E+03
C-17-f26	Ave	1.39E+03	5.28E+03	1.60E+03	4.40E+03	5.46E+03	**1.27E+03**	2.54E+03	1.94E+03	2.30E+03	1.57E+04
	Std	1.98E+03	2.03E+03	2.23E+03	1.82E+02	5.13E+02	**9.18E+01**	2.28E+02	2.23E+03	1.77E+02	8.19E+02
C-17-f27	Ave	**4.82E+02**	8.98E+02	6.29E+02	5.45E+02	1.17E+03	5.92E+02	6.00E+02	7.51E+02	7.48E+02	2.80E+03
	Std	1.11E+02	1.36E+02	5.61E+01	4.05E+01	1.25E+02	8.29E+01	**2.82E+01**	1.09E+02	7.01E+01	1.80E+02
C-17-f28	Ave	4.85E+02	5.04E+02	5.08E+02	**4.67E+02**	1.40E+03	4.94E+02	5.02E+02	5.12E+02	5.26E+02	1.03E+04
	Std	3.29E+01	2.40E+01	3.05E+01	1.85E+01	4.65E+02	2.24E+01	2.01E+01	3.21E+01	**1.78E+01**	6.44E+02
C-17-f29	Ave	1.54E+03	1.54E+03	8.61E+02	1.18E+03	1.53E+03	**3.60E+02**	6.58E+02	1.27E+03	1.12E+03	8.78E+03
	Std	2.15E+02	3.86E+02	2.33E+02	5.28E+02	3.29E+02	**2.23E+01**	1.05E+02	2.56E+02	2.31E+02	1.30E+03
C-17-f30	Ave	6.29E+05	9.35E+05	9.88E+05	**5.91E+05**	3.09E+06	5.96E+05	8.16E+05	9.06E+05	1.83E+06	1.31E+10
	Std	1.14E+05	1.49E+05	1.73E+05	2.40E+04	3.38E+06	**2.24E+04**	5.64E+04	1.24E+05	3.95E+05	3.91E+09

test suite, WSO and the other competitors' algorithms used a maximum number of FEs of $10000 \times d$ to conduct a fair comparison between them. The stop criterion for the algorithms was set to the maximum number of FEs, where each algorithm ran 51 times independently for each function in each experiment. The parameter settings of the optimization algorithms are presented in Table 5. For this test, the dimensions of the test functions considered in this work are 10, 30, 50 and 100. The mean error and standard deviation results of WSO and the other competing algorithms on the CEC-2017 benchmark functions with 10, 30, 50 and 100 dimensions are manifested in Tables 10, 11, 12 and 13, respectively. Specifically, these tables include the average values and standard deviations of error from the optimum solution of WSO and other nine state-of-the-art methods over 51 runs for each test function in the CEC-2017 set. The best results are put in bold in all of these tables for all test problems. The results of the other algorithms reported in these tables were taken from Ref. [1]. It is worth noting that the mean error and standard deviation values less than $1E-08$ are taken as zero [106].

When reading the mean error values of Tables 10, 11, 12 and 13, we can see that WSO reliably succeeded in solving the CEC-2017 test functions, where it reported optimal results for 4 functions in 10d, 2 functions in 30d, 2 functions in 50d and 2 functions in 100d out of a total of 29 test functions per each dimension. It is clearly observed that AMO is the best optimizer among all the algorithms, which gave the best results in 10d, 30d and 50d, while GSK is the best optimizer in test functions with 100d as can be seen from the results reported in Tables 10, 11, 12

Table 13
Optimization results of WSO and other algorithms over 51 independent runs on the CEC-2017 test functions of 100 variables with 100,0000 FEs.

F		WSO	TLBO	SFS	DE	GA	GSK	AMO	PSO	BBO	ACO
C-17-f1	Ave	7.72E+03	7.13E+03	8.57E+03	1.10E+04	2.03E+08	5.80E+03	**2.51E+03**	1.08E+04	1.28E+07	4.99E+11
	Std	2.13E+03	8.54E+03	1.21E+04	1.62E+04	1.62E+08	4.63E+03	**1.89E+03**	1.42E+04	5.83E+05	1.38E+11
C-17-f3	Ave	**2.84E+03**	6.30E+04	3.00E+04	4.23E+05	2.07E+05	1.15E+05	1.75E+05	3.17E+04	4.21E+05	1.32E+11
	Std	**4.36E+03**	1.28E+04	6.89E+03	2.81E+04	3.03E+04	2.15E+04	1.41E+04	5.19E+03	8.31E+04	2.81E+11
C-17-f4	Ave	2.16E+02	2.70E+02	2.72E+02	2.16E+02	4.76E+02	2.05E+02	**1.41E+02**	3.03E+02	2.87E+02	1.28E+05
	Std	5.50E+01	5.22E+01	3.84E+01	**2.36E+01**	6.04E+01	4.57E+01	6.04E+01	3.91E+01	4.43E+01	1.09E+04
C-17-f5	Ave	5.76E+02	5.90E+02	6.36E+02	8.19E+02	1.14E+03	5.31E+02	4.41E+02	6.02E+02	**2.24E+02**	1.98E+03
	Std	6.00E+01	5.32E+01	9.34E+01	**1.77E+01**	7.63E+01	3.39E+02	4.10E+01	6.13E+01	3.29E+01	3.89E+01
C-17-f6	Ave	1.33E+01	4.20E+01	1.76E+00	2.57E−03	4.24E+01	**1.72E−03**	6.28E−02	3.64E+01	8.79E−01	1.35E+02
	Std	4.56E+00	3.90E+00	1.94E+00	**4.31E−03**	4.85E+00	6.46E−03	8.93E−02	8.16E+00	4.51E−02	3.70E+00
C-17-f7	Ave	**2.88E+02**	1.35E+03	9.78E+02	9.39E+02	1.60E+03	8.75E+02	5.86E+02	5.71E+02	6.31E+02	8.86E+03
	Std	1.12E+02	2.09E+02	1.76E+02	1.95E+01	1.19E+02	**1.83E+01**	4.46E+01	1.17E+02	5.66E+01	1.95E+02
C-17-f8	Ave	6.63E+02	6.31E+02	6.27E+02	8.18E+02	1.08E+03	4.92E+02	4.33E+02	6.01E+02	**2.28E+02**	2.00E+03
	Std	1.28E+02	5.78E+01	9.44E+01	**2.02E+01**	9.95E+01	3.41E+02	4.24E+01	7.76E+01	3.48E+01	2.18E+01
C-17-f9	Ave	1.57E+04	2.82E+04	1.18E+04	**3.86E+00**	3.11E+04	8.46E+00	1.83E+03	1.81E+04	2.25E+03	1.25E+05
	Std	1.33E+03	9.68E+03	3.78E+03	4.56E+00	7.46E+03	**3.41E+00**	1.30E+03	3.09E+03	9.01E+02	7.21E+03
C-17-f10	Ave	**1.10E+04**	2.30E+04	1.21E+04	3.01E+04	2.28E+04	2.95E+04	1.81E+04	1.32E+04	1.10E+04	3.01E+04
	Std	1.46E+03	6.19E+03	1.33E+03	**3.96E+02**	1.51E+03	4.44E+02	7.17E+02	1.28E+03	9.31E+02	7.44E+02
C-17-f11	Ave	5.40E+02	1.16E+03	6.43E+02	6.05E+02	6.12E+04	**2.78E+02**	6.14E+02	1.12E+03	6.48E+04	1.17E+07
	Std	1.26E+02	2.76E+02	8.04E+01	8.70E+01	2.52E+04	**6.85E+01**	7.28E+01	2.00E+02	1.92E+04	3.79E+07
C-17-f12	Ave	7.11E+05	1.46E+06	2.11E+06	2.72E+05	8.67E+07	**8.34E+04**	1.34E+06	1.11E+07	4.10E+07	3.32E+11
	Std	2.98E+05	9.86E+05	9.27E+05	1.28E+05	3.86E+07	**7.52E+04**	5.48E+05	5.28E+06	1.51E+07	3.41E+10
C-17-f13	Ave	5.88E+03	8.96E+03	6.42E+03	5.32E+03	2.75E+06	3.20E+03	**2.37E+03**	4.22E+03	2.20E+06	8.01E+10
	Std	3.30E+03	4.36E+03	6.05E+03	4.39E+03	2.82E+06	2.64E+03	**1.03E+03**	3.95E+03	3.17E+05	1.35E+10
C-17-f14	Ave	1.48E+04	1.70E+05	**3.37E+02**	7.71E+03	9.93E+06	4.64E+03	8.50E+05	5.24E+05	1.54E+07	2.74E+08
	Std	4.93E+03	1.96E+05	**4.29E+01**	8.97E+03	5.46E+06	4.47E+03	3.54E+05	2.42E+05	7.21E+06	2.30E+08
C-17-f15	Ave	2.76E+03	2.41E+03	3.12E+03	8.91E+03	1.78E+06	7.33E+02	**5.84E+02**	2.17E+03	1.37E+06	3.65E+10
	Std	2.18E+03	2.13E+03	3.95E+03	7.19E+03	4.52E+05	1.09E+03	**3.50E+02**	1.59E+03	2.99E+05	5.65E+09
C-17-f16	Ave	3.41E+03	3.36E+03	3.35E+03	7.65E+03	6.22E+03	**2.27E+03**	3.28E+03	3.31E+03	3.44E+03	1.88E+04
	Std	6.04E+02	6.41E+02	6.00E+02	3.37E+02	7.26E+02	2.61E+03	**2.77E+02**	5.55E+02	7.75E+02	1.35E+03
C-17-f17	Ave	3.34E+03	3.17E+03	**2.30E+03**	4.71E+03	3.70E+03	3.91E+03	2.40E+03	2.93E+03	3.20E+03	4.34E+06
	Std	4.29E+02	5.95E+02	4.85E+02	4.85E+02	4.74E+02	6.68E+02	**2.29E+02**	5.54E+02	7.82E+02	2.67E+06
C-17-f18	Ave	3.85E+04	4.27E+05	**3.18E+04**	1.21E+05	8.40E+06	5.73E+04	1.60E+06	1.92E+06	6.60E+06	5.51E+08
	Std	**1.42E+04**	2.51E+05	2.32E+04	6.33E+04	4.08E+06	3.63E+04	4.67E+05	8.57E+05	3.01E+06	1.57E+08
C-17-f19	Ave	1.58E+03	2.36E+03	3.39E+03	8.95E+03	1.30E+06	**1.00E+03**	1.24E+03	1.93E+03	1.39E+06	3.35E+10
	Std	1.63E+03	2.03E+03	4.82E+03	1.05E+04	2.09E+05	**8.30E+02**	9.31E+02	2.77E+03	2.01E+05	8.97E+09
C-17-f20	Ave	2.39E+03	2.34E+03	**2.04E+03**	4.15E+03	3.92E+03	4.46E+03	2.30E+03	2.75E+03	2.74E+03	5.42E+03
	Std	5.65E+02	8.22E+02	4.66E+02	8.41E+02	4.42E+02	2.22E+02	2.63E+02	4.15E+02	4.91E+02	**1.66E+02**
C-17-f21	Ave	1.25E+03	8.80E+02	7.12E+02	1.04E+03	1.48E+03	6.07E+02	6.23E+02	9.70E+02	**4.78E+02**	2.62E+03
	Std	1.01E+02	8.83E+01	7.84E+01	2.54E+01	1.20E+02	3.37E+02	**2.31E+01**	1.34E+02	3.30E+01	6.65E+01
C-17-f22	Ave	1.57E+04	2.49E+04	1.30E+04	3.02E+04	2.41E+04	3.00E+04	1.94E+04	1.55E+04	**1.24E+04**	3.19E+04
	Std	2.45E+03	8.31E+03	5.94E+03	7.15E+02	1.07E+03	**4.57E+02**	7.43E+02	1.75E+03	1.12E+03	5.87E+02
C-17-f23	Ave	7.51E+02	1.35E+03	9.75E+02	8.60E+02	1.54E+03	**6.11E+02**	8.22E+02	1.49E+03	7.11E+02	3.91E+03
	Std	2.12E+02	1.05E+02	7.94E+01	2.99E+02	1.02E+02	**1.58E+01**	2.46E+0	1.49E+02	3.06E+01	1.67E+02
C-17-f24	Ave	2.39E+03	1.99E+03	1.54E+03	1.63E+03	2.11E+03	**9.32E+02**	1.23E+03	1.60E+03	1.20E+03	7.41E+03
	Std	3.16E+02	1.96E+02	1.00E+02	1.49E+02	1.33E+02	**1.70E+01**	4.73E+01	1.56E+02	4.17E+01	6.15E+02
C-17-f25	Ave	7.90E+02	8.28E+02	7.90E+02	**7.31E+02**	1.63E+03	8.21E+02	8.31E+02	8.12E+02	8.03E+02	4.90E+04
	Std	5.87E+01	6.11E+01	4.99E+01	5.56E+01	1.63E+02	**4.34E+01**	5.21E+01	7.29E+01	7.33E+01	5.90E+03
C-17-f26	Ave	8.18E+03	2.01E+04	1.37E+04	1.08E+04	1.58E+04	**3.66E+03**	7.91E+03	9.73E+03	6.26E+03	5.23E+04
	Std	1.05E+03	5.23E+03	6.56E+03	1.37E+03	1.46E+03	**1.78E+02**	6.30E+02	5.69E+03	4.03E+02	2.10E+03
C-17-f27	Ave	8.87E+02	1.42E+03	8.81E+02	**6.16E+02**	1.27E+03	6.57E+02	8.49E+02	9.31E+02	8.28E+02	7.92E+03
	Std	1.44E+02	2.02E+02	7.99E+01	**2.15E+01**	1.20E+02	3.07E+01	5.06E+01	1.14E+02	6.41E+01	2.58E+02
C-17-f28	Ave	5.64E+02	6.30E+02	6.13E+02	5.59E+02	2.60E+03	**5.53E+02**	5.60E+02	6.55E+02	6.44E+02	3.82E+04
	Std	8.74E+01	**2.91E+01**	3.09E+01	3.52E+01	1.62E+03	3.25E+01	3.26E+01	4.30E+01	4.08E+01	1.51E+03
C-17-f29	Ave	4.20E+03	4.34E+03	3.09E+03	4.74E+03	4.37E+03	**1.21E+03**	2.74E+03	3.66E+03	3.29E+03	1.19E+05
	Std	6.76E+02	6.73E+02	4.26E+02	1.19E+03	4.85E+02	**1.74E+02**	3.04E+02	5.24E+02	5.09E+02	5.53E+04
C-17-f30	Ave	1.19E+04	2.15E+04	1.22E+04	4.68E+03	3.17E+06	**2.99E+03**	5.25E+03	8.19E+03	2.63E+06	6.75E+10
	Std	7.74E+03	2.11E+04	6.93E+03	3.05E+03	3.02E+05	**2.73E+02**	1.28E+03	3.68E+03	4.88E+05	1.47E+10

and 13. Outstandingly, in optimizing test functions with 50d and 100d, it can be noted that WSO is the second best optimizer in 50d and the third best optimizer in 100d, where it scored optimal results similar to those reported by AMO and GSK in many test functions. Besides, WSO reported distinct mean error values that are better than those of other competing algorithms in 1 function in 10d, 2 functions in 30d, 5 functions in 50d and 4 functions in 100d. These findings bear out the power of WSO over others in optimizing quite challenging test functions such as those of high dimensions. From another aspect, when reading the standard deviation values of Table 10, WSO performed substantially better than many other well-known algorithms such as GAs and ACO by getting distinct standard deviation values in many test functions, and scored optimal values similar to those obtained by GSK and DE methods in the first 3 test functions. In Tables 11–13, we could read that WSO recorded better standard deviation values than those of other algorithms in 1, 2 and 2 functions in 30d, 50d and 100d, respectively. This underscores the fact that we have previously concluded that WSO has a high degree of stability when testing the functions under consideration in different search domains of different dimensions. In further details, WSO has remarkable performance in uni-modal test problems (C-17-f1, C-17-f3), where it is able to locate the global optimum solution continuously over 51 independent runs in these test functions in 10, 30, 50 and 100 dimensions. Also, WSO is capable of locating the optimum mean error solutions in three tests cases in the hybrid functions, namely C-17-15 and C-17-19 in 10d. Further, it got reasonable mean error values for the other problems, except for

Table 14
Optimization results of WSO over 25 independent runs on the CEC-2011 test functions with 150,000 FEs.

F	Best	Median	Ave	Worst	Std
C-11-f1	3.08E−27	8.41E+00	6.74E+00	1.73E+01	6.35E+00
C-11-f2	−2.84E+01	−2.65E+01	−2.69E+01	−2.53E+01	6.96E−01
C-11-f3	1.15E−05	1.15E−05	1.15E−05	1.15E−05	7.80E−20
C-11-f4	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
C-11-f5	−3.68E+01	−3.49E+01	−3.42E+01	−2.86E+01	2.56E+00
C-11-f6	−2.91E+01	−2.30E+01	−2.31E+01	−1.68E+01	3.37E+00
C-11-f7	6.04E−01	9.01E−01	9.20E−01	1.20E+00	1.70E−01
C-11-f8	2.20E+02	2.20E+02	2.20E+02	2.20E+02	0.00E+00
C-11-f9	0.00E+00	4.80E+03	2.96E+04	2.60E+05	6.54E+04
C-11-f10	−2.18E+01	−2.16E+01	−2.14E+01	−1.91E+01	5.03E−01
C-11-f11	5.15E+04	5.26E+04	5.26E+04	5.37E+04	5.23E+02
C-11-f12	1.07E+06	1.08E+06	1.09E+06	1.12E+06	1.16E+04
C-11-f13	1.54E+04	1.54E+04	1.54E+04	1.54E+04	1.21E+00
C-11-f14	1.80E+04	1.82E+04	1.82E+04	1.84E+04	9.12E+01
C-11-f15	3.28E+04	3.29E+04	3.29E+04	3.31E+04	8.25E+01
C-11-f16	1.29E+05	1.37E+05	1.37E+05	1.46E+05	4.50E+03
C-11-f17	1.89E+06	1.97E+06	2.11E+06	3.01E+06	2.84E+05
C-11-f18	9.44E+05	9.55E+05	1.19E+06	3.13E+06	5.59E+05
C-11-f19	9.99E+05	1.42E+06	1.50E+06	3.14E+06	4.54E+05
C-11-f20	9.42E+05	9.53E+05	1.05E+06	3.13E+06	4.37E+05
C-11-f21	1.03E+01	1.60E+01	1.59E+01	2.04E+01	2.65E+00
C-11-f22	1.52E+01	2.02E+01	2.06E+01	2.76E+01	2.56E+00

Table 15
Optimization results of WSO and other algorithms over 25 independent runs on the CEC-2011 test functions with 150,000 FEs.

F		WSO	TLBO	SFS	DE	GA	GSK	AMO	PSO	BBO	ACO
C-11-f1	Ave	6.74E+00	6.60E+00	4.69E+00	4.30E+00	1.95E+01	3.28E+00	**4.85E−01**	2.63E+01	2.05E+01	3.05E+01
	Std	6.35E+00	6.78E+00	4.49E+00	5.38E+00	3.01E+00	5.21E+00	1.17E+00	**0.00E+00**	3.40E+00	8.23E−01
C-11-f2	Ave	**−2.69E+01**	−2.08E+01	−2.66E+01	−1.31E+01	−9.87E+00	−1.13E+01	−1.98E+01	−3.79E+00	−1.85E+01	−6.16E+00
	Std	6.96E−01	2.15E+00	1.28E+00	4.60E+00	2.20E+00	1.03E+00	1.03E+00	**5.52E−02**	1.85E+00	8.28E−01
C-11-f3	Ave	**1.15E−05**	**1.15E−05**	**1.15E−05**	**1.15E−05**	**1.15E−05**	**1.15E−05**	**1.15E−05**	**1.15E−05**	**1.15E−05**	**1.15E−05**
	Std	7.80E−20	4.08E−17	4.87E−10	1.86E−15	9.88E−10	9.12E−13	7.25E−14	1.46E−12	3.80E−09	**0.00E+00**
C-11-f4	Ave	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Std	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
C-11-f5	Ave	**−3.42E+01**	−2.83E+01	−3.37E+01	−1.95E+01	−1.38E+01	−2.06E+01	−3.32E+01	−1.98E+01	−3.30E+01	−1.48E+01
	Std	2.56E+00	3.70E+00	1.61E+00	9.34E−01	1.94E+00	1.21E+00	**7.33E−01**	9.91E−01	7.34E−01	1.21E+00
C-11-f6	Ave	−2.31E+01	−1.94E+01	−2.74E+01	−1.41E+01	−3.35E+00	−6.94E+00	−2.70E+01	−1.06E+01	**−2.79E+01**	−1.04E+01
	Std	3.37E+00	1.70E+00	1.86E+00	1.53E+00	3.99E+00	2.48E+00	9.45E−01	1.03E+00	**7.95E−01**	1.42E+00
C-11-f7	Ave	**9.20E−01**	1.16E+00	1.36E+00	1.75E+00	1.13E+00	1.78E+00	1.37E+00	1.53E+00	1.44E+00	1.89E+00
	Std	1.70E−01	2.69E−01	1.39E−01	9.79E−02	1.61E−01	1.08E−01	9.38E−02	1.21E−01	**8.64E−02**	9.89E−02
C-11-f8	Ave	**2.20E+02**	**2.20E+02**	**2.20E+02**	**2.20E+02**	**2.20E+02**	**2.20E+02**	**2.20E+02**	2.21E+02	**2.20E+02**	2.39E+03
	Std	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.84E+00	**0.00E+00**	1.32E+03
C-11-f9	Ave	2.96E+04	3.99E+03	2.27E+04	2.37E+04	1.32E+05	2.11E+03	**1.15E+03**	1.85E+06	7.83E+04	1.03E+06
	Std	6.54E+04	2.88E+03	6.55E+03	2.35E+03	3.76E+04	5.02E+02	**3.24E+02**	1.54E+05	2.42E+04	2.42E+04
C-11-f10	Ave	−2.14E+01	−1.89E+01	−2.15E+01	−1.41E+01	−9.61E+00	**−2.16E+01**	−2.13E+01	−9.31E+00	−1.52E+01	−9.04E+00
	Std	5.03E−01	2.43E+00	1.31E−01	2.68E+00	1.67E+00	1.19E−01	**9.12E−02**	8.20E−01	1.78E+00	1.49E−01
C-11-f11	Ave	5.26E+04	3.79E+06	4.92E+05	1.26E+05	5.52E+05	**5.24E+04**	5.26E+04	5.36E+06	5.66E+04	9.25E+06
	Std	5.23E+02	9.41E+05	2.14E+05	2.43E+04	8.33E+05	6.88E+02	**5.02E+02**	3.85E+05	8.38E+02	2.73E+06
C-11-f12	Ave	1.09E+06	1.53E+06	1.30E+06	1.14E+06	1.05E+07	**1.07E+06**	**1.07E+06**	1.42E+07	1.09E+06	8.10E+06
	Std	1.16E+04	2.23E+05	6.45E+04	9.67E+03	7.73E+05	1.73E+03	**1.27E+03**	9.63E+05	1.93E+04	2.33E+05
C-11-f13	Ave	**1.54E+04**	1.55E+04	**1.54E+04**	**1.54E+04**	1.55E+04	**1.54E+04**	**1.54E+04**	1.56E+04	1.55E+04	1.29E+05
	Std	**1.21E+00**	1.24E+01	1.44E+00	1.18E+01	2.23E+01	2.44E+00	2.59E+00	5.13E+01	2.49E+01	9.70E+04
C-11-f14	Ave	**1.82E+04**	1.93E+04	1.88E+04	1.84E+04	1.98E+04	1.84E+04	1.92E+04	1.97E+04	1.94E+04	4.15E+05
	Std	9.12E+01	9.32E+01	**8.04E+01**	1.66E+02	7.01E+02	1.22E+02	1.48E+02	2.14E+02	3.21E+02	4.79E+05
C-17-f15	Ave	3.29E+04	3.29E+04	3.30E+04	3.29E+04	3.30E+04	**3.28E+04**	3.30E+04	1.26E+05	3.31E+04	4.13E+06
	Std	8.25E+01	7.14E+01	2.35E+01	3.03E+01	7.93E+01	**1.55E+01**	2.09E+01	5.23E+04	6.90E+01	3.30E+06
C-11-f16	Ave	1.37E+05	1.36E+05	1.37E+05	1.37E+05	1.50E+05	**1.35E+05**	1.37E+05	4.76E+06	1.42E+05	7.43E+07
	Std	4.50E+03	3.40E+03	2.16E+03	2.91E+03	7.18E+03	2.22E+03	**1.70E+03**	4.24E+06	4.91E+03	1.77E+07
C-11-f17	Ave	2.11E+06	2.05E+06	2.21E+06	2.26E+06	8.78E+08	2.09E+06	**2.02E+06**	1.24E+10	2.70E+06	1.78E+10
	Std	2.84E+05	2.21E+05	2.60E+05	2.34E+05	1.10E+09	**1.20E+05**	1.63E+05	2.39E+09	1.98E+06	3.18E+09
C-11-f18	Ave	1.19E+06	1.14E+06	1.05E+06	1.88E+06	1.32E+06	1.27E+06	1.04E+06	1.29E+08	**9.70E+05**	1.53E+08
	Std	5.59E+05	8.67E+04	4.38E+04	3.06E+05	2.70E+05	7.56E+04	7.56E+04	1.15E+07	**1.47E+04**	1.89E+07
C-11-f19	Ave	1.50E+06	**1.42E+06**	1.51E+06	2.52E+06	2.24E+06	2.00E+06	1.56E+06	1.34E+08	1.53E+06	1.47E+08
	Std	4.54E+05	1.58E+05	1.93E+05	2.58E+05	2.48E+06	1.36E+05	**1.29E+05**	1.95E+07	2.56E+05	2.71E+07
C-11-f20	Ave	1.05E+06	1.12E+06	1.05E+06	1.77E+06	1.37E+06	1.29E+06	1.03E+06	1.37E+08	**9.75E+05**	1.59E+08
	Std	4.37E+05	8.06E+04	4.64E+04	2.78E+05	1.09E+06	9.20E+04	5.94E+04	1.98E+07	**1.54E+04**	1.44E+07
C-11-f21	Ave	1.59E+01	**1.55E+01**	1.77E+01	1.77E+01	3.15E+01	1.70E+01	1.85E+01	6.11E+01	2.24E+01	8.61E+01
	Std	2.65E+00	2.02E+00	3.13E+00	3.62E+00	6.07E+00	3.11E+00	**1.39E+00**	5.34E+00	3.32E+00	2.33E+01
C-11-f22	Ave	2.06E+01	2.12E+01	2.00E+01	1.32E+01	3.50E+01	**1.29E+01**	2.22E+01	5.09E+01	2.52E+01	8.85E+01
	Std	2.56E+00	2.41E+00	3.09E+00	2.78E+00	8.81E+00	2.93E+00	**1.81E+00**	4.65E+00	2.77E+00	1.26E+01

Table 16
Mean fitness values of WSO using various values for the parameter a_0.

a_0	Functions				
	C-17-f1 (10d)	C-17-f1 (30d)	C-17-f1 (50d)	C-17-f1 (100d)	C-11-f1
2	0.00E+00	4.65E+01	4.15E+03	8.98E+05	2.76E+01
5	0.00E+00	2.45E−01	5.29E+01	3.18E+03	8.89E+00
10	0.00E+00	4.89E−02	8.27E+02	2.87E+04	1.51E+01
15	0.00E+00	5.08E−02	9.67E+02	9.48E+04	5.76E+01

Table 17
Mean fitness values of WSO using different values for the parameter a_1.

a_1	Functions				
	C-17-f1 (10d)	C-17-f1 (30d)	C-17-f1 (50d)	C-17-f1 (100d)	C-11-f1
1	2.13E−05	8.17E+03	6.17E+04	1.18E+06	2.27E+02
50	7.29E−07	9.43E+02	4.13E+03	6.39E+05	5.63E+01
100	0.00E+00	6.39E−04	1.07E+02	8.38E+03	9.23E+00
150	7.45E−08	1.79E+03	5.28E+03	2.76E+04	5.82E+01

Table 18
Mean fitness values of WSO using different values for the parameter a_2.

a_2	Functions				
	C-17-f1 (10d)	C-17-f1 (30d)	C-17-f1 (50d)	C-17-f1 (100d)	C-11-f1
0.05	1.35EE-06	2.65E+04	6.78E+04	2.38E+06	9.79E+01
0.005	2.39EE-07	3.67E+02	3.84E+03	3.64E+05	3.37E+01
0.0005	0.00E+00	9.44E−04	1.18E+02	6.84E+03	9.65E+00
0.00005	6.89E−08	4.18E+02	5.81E+03	9.45E+03	2.59E+01

Table 19
Mean fitness values of WSO using different values for the parameter τ.

τ	Functions				
	C-17-f1 (10d)	C-17-f1 (30d)	C-17-f1 (50d)	C-17-f1 (100d)	C-11-f1
4.11	0.00E+00	5.16E−04	6.45E+01	7.33E+04	2.34E+01
4.12	0.00E+00	6.53E−03	7.31E+02	6.53E+03	3.81E+01
4.125	0.00E+00	8.49E−05	5.57E+01	9.87E+02	5.89E+00
4.13	6.28E−07	5.13E−02	4.83E+03	7.72E+05	8.54E+01

Table 20
Average ranking of WSO alongside other algorithms using Friedman's test based upon their results on the CEC-2017 test functions, each with dimensions of 10.

Algorithm	Rank
WSO	3.96551724
TLBO	5.22413793
SFS	3.86206896
DE	3.34482758
GA	8.65517241
GSK	3.99999999
AMO	3.15517241
PSO	6.258620689
BBO	7.24137931
ACO	9.29310344

C-17-12, where WSO did not perform very well in this test case. Regarding the composition functions, which are the most difficult ones in the CEC-2017, the optimal solutions were detected by WSO in C-17-23, C-17-25 and C-17-26 test cases in 10d. However, it failed to get the optimal solutions in some test cases such as in C-17-12 and C-17-30 in dimensions 30, 50 and 100. As a result, WSO is sometimes trapped in a local optimal, but it is not far from optimal in all test functions of the CEC-2017. Lastly, it can be seen that TLBO, PSO and BBO algorithms delivered reasonable solutions in different test functions of CEC-2017 in some of the dimensions considered. However, GA and ACO behaved poorly in all of the test functions in all dimensions. In general, AMO, GSK, WSO, SFS and DE acted much better than the others in most of the test functions in all the dimensions considered. Besides, it can be shown that WSO competes strongly with AMO, GSK and SFS in many test functions in 10, 30, 50 and 100 dimensions, and outperformed others in the majority of test functions in all dimensions. This attainment is further confirmation of the ability of WSO to outperform well-studied and recent meta-heuristics while also reporting highly competitive outcomes with high performance optimizers such as AMO,GSK, SFS and DE in broadly well-known benchmark test functions. In sum, the performance of competing algorithms such as BBO, PSO, GA, ACO, and TLBO presents severe degradation as the search space dimensions grow from 10d to 100d, while the overall performance of WSO is almost stable at that and moderately dwindles.

4.4. Convergence analysis of WSO on CEC-2017

Convergence analysis is an essential feature to better understand the exploration and exploitation behaviors of optimization algorithms. In light of this, to study the convergence behavior of the proposed WSO, the convergence curves of WSO in terms of the fitness values, represented in Log_{10}, of the median run over 51 runs for each test problem of the CEC-2017 benchmark problems with dimensions 10, 30, 50 and 100 are presented in Fig. A.1. The fitness values of the convergence curves in Fig. A.1 denote the best solutions got so far in terms of function evaluations of the optimization process of the proposed WSO. In Fig. A.1, the y-axis which represents the fitness value is represented by Log_{10}, while the x-axis is represented by the number of function evaluations. Additionally, for all of the convergence graphs in Fig. A.1, the population size used by WSO is 100 correlated with a maximum number of FEs of $10000 \times d$, where d stands for the dimensions

Table 21
Results of Holm's test for the CEC-2017 test group with dimensions of 10 for each function.

i	WSO vs.	z-value	p-value	α/i (0.05)	Hypothesis
9	ACO	7.71969755	1.16606054E−14	0.00555555	Rejected
8	GA	6.91736944	4.60107080E−12	0.00625	Rejected
7	BBO	5.13923685	2.75856500E−07	0.00714285	Rejected
6	PSO	3.90321786	9.49221853E−05	0.00833333	Rejected
5	TLBO	2.60214524	0.00926426	0.01	Rejected
4	GSK	1.06254264	0.28798941	0.0125	Not rejected
3	WSO	1.01917355	0.30812057	0.01666666	Not rejected
2	SFS	0.88906629	0.37396745	0.025	Not rejected
1	DE	0.23852998	0.81147006	0.05	Not rejected

Table 22
Average ranking of WSO and other algorithms using Friedman's test on their results on the CEC-2017 test functions, each with dimensions of 30.

Algorithm	Rank
WSO	4.06896551
TLBO	5.68965517
SFS	3.46551724
DE	4.15517241
GA	8.793103448
GSK	3.31034482
AMO	3.13793103
PSO	6.27586206
BBO	6.10344827
ACO	10.00000000

of the problems. More specifically, in Fig. A.1, the total number of function evaluations used by the proposed WSO are 100,000, 300,000, 500,000 and 100,0000 for problems of dimensions 10, 30, 50 and 100, respectively.

In the analysis of the convergence curves in Fig. A.1, it is evident from the convergence graphs in this figure that the convergence speed of the proposed WSO is rapid in the first few evaluations of the optimization process for the test problems of the CEC-2017. As shown in Fig. A.1, there are different behaviors of the WSO's convergence curves during its iterative process for different test functions. It is noticeable that the convergence speed of WSO decreases significantly, with marked improvement in the half and final evaluations of the optimization process. This is due to the outstanding global search strategy in the initial stage of WSO as well as the local search for the best preserved position of the white sharks in the search space. However, these convergence curves encounter little sudden glitches in the first few evaluations of the optimization process. Then, they progressively converge after the first few evaluations to exploit the near-global or global optimal solutions in order to achieve the optimal response in the final evaluations. This indicates that the white sharks in WSO switch abruptly at the outset and resort to fluctuations proportional to the number of function evaluations. According to this view, WSO first requires the white sharks to revolve around the search space and cause a large exploration of the search space. Then, WSO exploits the search space by inspiring its search agents to the global optimum and stimulating them to search in local areas of the search space. In addition, the convergence curves in Fig. A.1 indicate that WSO can obtain plausible solutions in most of the CEC-2017 test problems in a smaller number of generations below the maximum predefined number of FEs. Overall, WSO is sufficiently scalable and can substantially balance exploration and exploitation capabilities so that the maximum number of FEs is achieved. Finally, as can be obviously deduced from Fig. A.1, WSO proved to be an efficient and powerful method for solving unconstrained optimization functions within a finite number of FEs. This is a quite important matter for WSO when tackling complex real-world optimization problems.

4.5. Performance of WSO on IEEE CEC-2011

The CEC-2011 test suite that contains 22 challenging and up-to-date test benchmark functions was used to provide an additional challenge to the performance of WSO on real-world problems. This is to further test the competency and reliability scores of the proposed WSO. These test problems implicate a large number of local optima and different shapes of test functions in various regions and dimensions. Therefore, they were used here to assess the local optimal avoidance accuracy and exploration capacity of WSO. More details about CEC-2011 can be found in [107]. For each of the test functions in the CEC-2011, the proposed WSO used a maximum number of FEs of 150,000 associated with 100 white sharks (i.e., population size). The parameter settings of WSO applied to optimize these test functions are shown in Table 5. Table 14 summarizes the statistical outcomes of WSO on the CEC-2011 test functions. These results are presented in terms of the best, median, average, worst and standard deviation values of the objective function values of WSO over 25 independent runs for all 22 benchmark test functions.

As it is clearly exhibited in Table 14, WSO is capable of finding the global optimum solution constantly in two test cases. While WSO did not consistently find optimum solutions in the first benchmark test function, the best achieved outcome is roughly close to the global optimum solution that can be substantiated by a small standard deviation value. In addition, for the remaining benchmark test functions, the margin differences between the mean and median results are small even in cases when the final outcomes are far away from the optimum results. These findings confirm that WSO is effective in most of the CEC-2011 test functions. Additionally, the standard deviations of WSO in these test functions are not large, which denotes that the superiority of WSO is rooted.

Finally, since no precise optimal performance is available for the majority of these test cases, it is tough to estimate the absolute performance level of WSO at this point. In this, the relative performance of WSO can be evaluated and obtained when the performance of WSO is compared to that of other competing algorithms. In this set, the performance of WSO was compared with that of 9 existing optimization algorithms that have scored promising performance in the literature and shown in Table 5. This comparison was performed to give a thorough study of the accuracy of WSO in optimization. For WSO and each algorithm evaluated, the maximum number of FEs for all of the 22 test functions was set to 150,000, considering that each algorithm was carried out using a population of 50 or 100 search agents, as given in Table 5, correlated with 25 independent runs in all experiments. The number of FEs was used as a termination method to stop each algorithm in order to make a fair comparison between WSO and those comparative algorithms. The performance results of the comparisons of WSO with other algorithms in terms of mean errors and standard deviations of CEC-2011 test functions are summarized in Table 15. The best results are highlighted in

Table 23
Results of Holm's test for the CEC-2017 test functions with dimensions of 30 for each function.

i	WSO vs.	z-value	p-value	α/i (0.05)	Hypothesis
9	ACO	8.63044839	6.11119199E−18	0.00555555	Rejected
8	GA	7.11253033	1.13934481E−12	0.00625	Rejected
7	PSO	3.94658695	7.92731399E−05	0.00714285	Rejected
6	BBO	3.72974151	1.91676302E−04	0.00833333	Rejected
5	TLBO	3.20931246	0.00133052	0.01	Rejected
4	DE	1.27938807	0.20076042	0.0125	Not rejected
3	WSO	1.17096536	0.24161270	0.01666666	Not rejected
2	SFS	0.41200633	0.68033478	0.025	Not rejected
1	GSK	0.21684543	0.82832880	0.05	Not rejected

Table 24
Average ranking of WSO and other algorithms using Friedman's test based upon their results on the CEC-2017 test functions, each with dimensions of 50.

Algorithm	Rank
WSO	3.77586206
TLBO	5.56896551
SFS	3.89655172
DE	4.49999999
GA	8.44827586
GSK	3.77586206
AMO	3.48275862
PSO	5.82758620
BBO	5.72413793
ACO	10.0

bold for all test problems in Table 15, where the results of other algorithms are obtained from Ref. [1]. It can be seen that all algorithms were compared with equal floating point precision in order not to bias any algorithm over others, so the margin differences between their outcomes can be traced back to their performance obtained.

It can be seen from Table 15 that WSO is adept in consistently finding the global optimal solution in 3 test cases over 25 independent runs. Although the optimal solutions were not constantly found for all test cases of the CEC-2011 benchmark set, the findings of WSO are very near to the global optimal solution which can be demonstrated by the small standard deviation values obtained. In it, WSO reported distinct mean error values better than those reported by other rival competing algorithms in 4, and got optimal results similar to other promising algorithms such as GSK, AMO and SFS in 4 out of 22 test functions. These results point out that WSO is superior to other competing algorithms in some of the CEC-2011 test functions that presented itself as the second best optimizer after AMO. The third and fourth best optimizers are GSK and SFS, respectively, where both scored optimal results in 4 test cases and GSK reported best results in other 5 test cases. The difference margins between the mean error and standard deviation results got by WSO and those obtained by the other best competitive optimizers such as GSK and AMO are very slight and statistically non-significant. In regards to the standard deviations presented in Table 15, WSO behaved well by achieving distinct standard deviation values in several test functions, and fulfilling optimal standard deviations analogous to GSK, AMO and SFS in other 2 test functions. These results indicate that the superiority of WSO is well rooted. From Table 15, it can be noticed that TLBO and BBO are excellent in optimizing some test functions of the CEC-2011 benchmark, where DE also showed sensible performance compared to others. However, GA, ACO and PSO are not good in optimizing most of the CEC-2011 test functions. Overall, it can be revealed that WSO, GSK, AMO and SFS obtained promising optimization results in most of the CEC-2011 test functions.

From the above findings and comparisons, the overall performance of WSO in solving small, medium and high dimensional optimization problems is so encouraging compared to its counterpart algorithms. This indicates that WSO is stable, efficient and robust with excellent exploration and exploitation behaviors.

4.6. Convergence analysis of WSO on CEC-2011

To further study the convergence behavior of WSO on other complex optimization problems, referred to as CEC-2011, the convergence curves of WSO, in terms of the best fitness values of the median run, over a number of 25 runs, for all test problems of the CEC-2011 benchmark test suite, were obtained as presented in Fig. A.2. The fitness values of the convergence curves in Fig. A.2 indicate the best solutions obtained so far in terms of FEs of the optimization process of the proposed WSO. In the convergence graphs of Fig. A.2, the y-axis implements the fitness values obtained so far, while the x-axis is represented by the number of FEs. In addition, for all of the convergence curves in Fig. A.2, the population size of WSO is 100 associated with a maximum number of FEs of 150,000. It is known that the required number of iterations is equal to the maximum number of function evaluations/population size, so the number of iterations is 1500. It is obvious from the convergence results of Fig. A.2 that WSO delivered reasonable convergence behaviors for most of the test problems. This is due to that WSO converged to the optimum solutions during the optimization process. These curves demonstrate that WSO converges rapidly towards the global optimum solutions, and thus possesses robust convergence ability. Furthermore, WSO shows faster convergence response in some test problems such as C-11-f3 to C-11-f4. Finally, the convergence curves shown in Fig. A.2 reveal that WSO maintains a satisfactory balance between exploration and exploitation to effectively find the global optimum solution. In short, WSO achieved promising performance even for complex test problems with high dimensions. This confirms that WSO is able to effectively exploring the search space, avoid local optimal solutions, and converge somewhat towards optimality.

4.7. Sensitivity analysis of WSO's parameters

This section studies the sensitivity of WSO to: (1) the control parameter a_0, (2) the control parameter a_1, (3) the control parameter a_2, and (4) the control parameter τ. This analysis is useful for determining which of these parameters are robust and sensitive to various input values, and which ones have a large impact on the accuracy of WSO. This study carried out a full design for WSO with these parameters on some functions selected from the CEC-2017 and CEC-2011 test suites. These functions are: C-17-f1 (10d), C-17-f1 (30d), C-17-f1 (50d), C-17-f1 (100d) and C-11-f1. The values of each parameter were determined to design a sensitivity analysis as described below. In performing the sensitivity analysis procedure below, WSO employed 100 white sharks associated with a maximum number of FEs of $10,000 \times d$ for C-17-f1 and a maximum number of 150,000 FEs for C-11-f1. The sensitivity analysis results of WSO for C-17-f1 in all considered dimensions

Table 25
Results of Holm's test for the CEC-2017 test functions with dimensions of 50 for each function.

i	WSO vs.	z-value	p-value	α/i (0.05)	Hypothesis
9	ACO	8.19675752	2.46957891E−16	0.00555555	Rejected
8	GA	6.24514858	4.23397424E−10	0.00625	Rejected
7	PSO	2.94909794	0.00318702	0.00714285	Rejected
6	BBO	2.81899068	0.00481749	0.00833333	Rejected
5	TLBO	2.62382978	0.00869472	0.01	Rejected
4	DE	1.27938807	0.20076042	0.0125	Not rejected
3	SFS	0.52042904	0.60276456	0.01666666	Not rejected
2	GSK	0.36863724	0.71239813	0.025	Not rejected
1	WSO	0.36863724	0.71239813	0.05	Not rejected

Table 26
Average ranking of WSO and other algorithms using Friedman's test based upon their results on the CEC-2017 test functions, each with dimensions of 100.

Algorithm	Rank
WSO	4.36206896
TLBO	6.03448275
SFS	4.36206896
DE	5.37931034
GA	8.27586206
GSK	2.93103448
AMO	3.24137931
PSO	5.34482758
BBO	5.12068965
ACO	9.94827586

and C-11-f1 were studied in terms of the mean fitness over 51 independent runs and 25 independent runs, respectively, as presented below:

1. Control parameter a_0: to verify the effect of a_0 on the efficiency of WSO, it was carried out for several values of a_0 taken as 2, 5, 10 and 15, as shown in Table 16, while keeping other parameters unchanged.
   Table 16 shows that WSO has a relatively good sensitivity to this parameter, providing relatively reasonable differences between the outcomes. However, WSO's behavior remains stable, as WSO can still get promising performance results.
2. Control parameter a_1: to check the sensitivity of WSO to a_1, it was simulated for different values of this parameter such as 1, 50, 100 and 150, where the other parameters were unaltered. Table 17 illustrates the effect of this parameter on the mean fitness of WSO.
   It is apparent from Table 17 that WSO is robustly sensitive to a_1, as there is a big difference between the results based on the values of this parameter. Obviously, WSO yielded the best results when a_1 is 100.
3. Control parameter a_2: Table 18 shows the mean fitness values of WSO in solving C-17-f1 and C-11-f1 with different values of a_2, while keeping the other control parameters unchanged.
   It is evident from Table 18 that WSO is sensitive to a_2, as there is a relatively fair difference between the results got using different values of this parameter. It is seen from Table 18 that WSO achieved the best results when the value of a_2 was set to 0.0005. The points out that a good balance between exploration and exploitation behaviors could be achieved with this value.
4. Control parameter τ: to explore the sensitivity of WSO to τ, it was simulated for different values of this parameter as illustrated in Table 19.
   The results in Table 19 show that WSO can reach the best solution when the value of τ is between 4.11 and 4.12, although the results are reasonable when τ is 4.13. This props up the importance of fine-tuning this parameter on the strength of WSO.

Generally, from the results presented in Tables 16, 17, 18 and 19, it is clear that there is a large need to study the sensitivity of the above-mentioned control parameters on the performance of WSO. Furthermore, there is a need to well-adjust these parameters to help WSO to get the optimum global solutions.

4.8. Statistical test analysis

The substantive statistical analysis using the mean and standard deviation score metrics of the results gives an in-depth understanding of the accuracy and stability of WSO. These tests revealed cogent exploration and exploitation capabilities of WSO, but they are incapable of proving how good the WSO is. Therefore, other statistical tests are needed to ensure that the results of WSO are not typically generated fortuitously. Statistical tests are essential to study the consistency and performance of WSO through the use of the mean values achieved in all of independent runs performed.

In this section, two statistical tests were conducted to order the optimization methods and to grasp whether the differences in terms of quality between all methods are statistically significant. The first test performed in this work was Friedman's statistical test [112]. To realize a dependable comparison by this test, a comparison of more than 10 test functions is required, whereby the comparison must be made between more than 5 different methods [113]. In this regard, this study compared between more than 5 different methods, each of which was tested on 51 functions. These functions are grouped into two test bunches, as follows: (1) the first test set is the CEC-2017 with 29 test functions, and (3) the second test set is the CEC-2011 which consists of 22 test cases.

To rank the algorithms using Friedman's test, there is a need to calculate the average ranked value. A comparison is then made to assess the critical values obtained for the significance level (p-value) which was defined to be $\alpha = 0.05$. In the event that the significance level divulged by Friedman's test is less than or equal to 0.05, the null hypothesis is rejected, indicating the there is no difference between the accuracy of all the compared methods. The alternative hypothesis asserts that there is a difference between the performances of all the compared methods. When applying Friedman's test, the best algorithm is the one that receives the lowest rank while the worst algorithm receives the highest rank. The best algorithm is used as a control method for subsequent analysis. In order to assess the statistical performance of WSO and each other method on the CEC-2017 test suite, the mean error values of all methods in all dimensions and the rank of each method in each dimension were taken into account. The average rankings of WSO in conjunction with other methods using Friedman's test on CEC-2017 with a dimension of 10 for all functions in this group are summarized in Table 20.

Table 27
Results of Holm's test for the CEC-2017 test functions with dimensions of 100 for each test function.

i	WSO vs.	z-value	p-value	α/i (0.05)	Hypothesis
9	ACO	8.82560928	1.08865472E−18	0.0055555	Rejected
8	GA	6.72220854	1.78990515E−11	0.00625	Rejected
7	TLBO	3.90321786	9.49221853E−05	0.00714285	Rejected
6	DE	3.07920520	0.00207553	0.00833333	Rejected
5	PSO	3.03583611	0.00239869	0.01	Rejected
4	BBO	2.75393705	0.00588830	0.0125	Rejected
3	WSO	1.79981712	0.07188951	0.01666666	Not rejected
2	SFS	1.79981712	0.07188951	0.025	Not rejected
1	AMO	0.39032178	0.69629861	0.05	Not rejected

Table 28
Average ranking of WSO and other algorithms using Friedman's test based upon their results on the CEC-2011 test functions.

Algorithm	Rank
WSO	3.74999999
TLBO	4.34090909
SFS	3.90909090
DE	5.31818181
GA	7.20454545
GSK	3.88636363
AMO	3.63636363
PSO	8.45454545
BBO	5.22727272
ACO	9.27272727

In Table 20, the p-value got by the application of Friedman's test on CEC-2017 with dimension 10 is 1.03360431E−10. This indicates that there are statistically major differences in performance between the competing algorithms. As per the results in Table 20, AMO is the control algorithm. In this table, WSO is ranked fourth after AMO, DE and SFS, but the differences between WSO and those algorithms are small, with WSO followed in order by: GSK, TLBO, PSO, BBO, GA, and ACO.

More steps are needed to figure out which algorithms perform significantly different from WSO and which ones are alike. These steps are important to verify whether the performance of WSO statistically deviates from that of other algorithms. To this end, a post-hoc statistical method, referred to as Holm's test [114] was considered. This method is important for finding out which methods are worse or better than WSO. Holm's test sorts all algorithms based on their p-values and compares them to $\alpha/k-i$, where k is the freedom degree and i is the method number. This method starts with the most important p-value and consecutively rejects the null hypothesis under the condition that $p_i < \alpha/k-i$. Once the method is under to reject the hypothesis, it ceases and all remaining hypotheses are deemed acceptable.

The results of applying Holm's test, as a post-hoc method after Friedman's test, on the CEC-2017 test group with a dimension of 10 are shown in Table 21.

In Table 21, hypotheses that have a p-value $\leq$ 0.0125 are rejected based on Holm's test. As can be inferred from this table, the performance of WSO is statistically significant and better than some of the competing methods mentioned above, with little difference from those better algorithms.

Table 22 provides a summary of the ranking results generated by Friedman's test when applied to the mean error results of the CEC-2017 benchmark with dimensions 30 for all functions.

The p-value revealed by Friedman's test on the results of the CEC-2017 functions with a dimension of 30 is 9.34508026E−11. This p-value indicates that there is a statistically large difference between the algorithms, which means that the null hypothesis is rejected. It is evidently seen from the results shown in Table 22 that WSO achieved a reasonable score among all other competitors. The rank of the algorithms in Table 22 are in succession as follows: AMO, GSK, SFS, WSO, DE, TLBO, BBO, PSO, GA, ACO. Looking at Table 22 once more, we can observe that the difference between WSO and AMO, GSK and SFS is relatively small, while the difference between WSO and DE, TLBO, BBO, PSO, GA, and ACO is large.

The results obtained by Holm's method based on the results of Table 22 are shown in Table 23.

In Table 23, the hypotheses with p-value $\leq$ 0.0125 were rejected by the Holm's test method. The results in this table confirm that WSO has a good standing among its top rivals, where this result is consistent with previous ones.

Table 24 reveals the statistical results of applying Friedman's test to the results of WSO and other competing algorithms on the CEC-2017 benchmark test set in 50d test problems.

In Table 24, the p-value revealed by Friedman's test for the mean error values of the CEC-2017 functions with dimensions of 50 is 9.99175187E−11. The null hypothesis was rejected as per these results, which implies that there are statistically big differences between the rival algorithms. In Table 24, AMO outperformed the proposed WSO and all other algorithms, with the lowest rank reported for AMO is 3.48275862 followed by GSK and WSO in the same rank, SFS, DE, TLBO, BBO, PSO, GA and ACO in the last rank. Table 25 shows the analysis results of the application of Holm's test as a subsequent method after Friedman's test for the CEC-2017 set with 50d.

In Table 25, the hypotheses with p-value $\leq$ 0.0125 were rejected by the application of the Holm's test method. The results in this table show that AMO is statistically the best one followed by GSK and WSO in the same rank.

A summary of the statistical analysis results of applying Friedman's on the CEC-2017 test suite with 100d test functions is given in Table 26.

According to Friedman's test at $\alpha = 0.05$, a significant difference can be inferred from the results reported in Table 26. Instead, to be more precise, the p-value got by Friedman's test on the results of CEC-2017 in 100d is 1.02187369E−10. It is clear from Table 26 that GSK outperforms other algorithms in these test cases with the considered dimension. More specifically, we can observe from Table 26 that WSO is one of the best optimization algorithms, which has the same rank as SFS and both come after GSK and AMO. In short, the order of the algorithms with respect to their superiority is as follows: GSK, AMO, WSO, SFS, BBO, PSO, DE, TLBO, GA, and in the last place comes ACO.

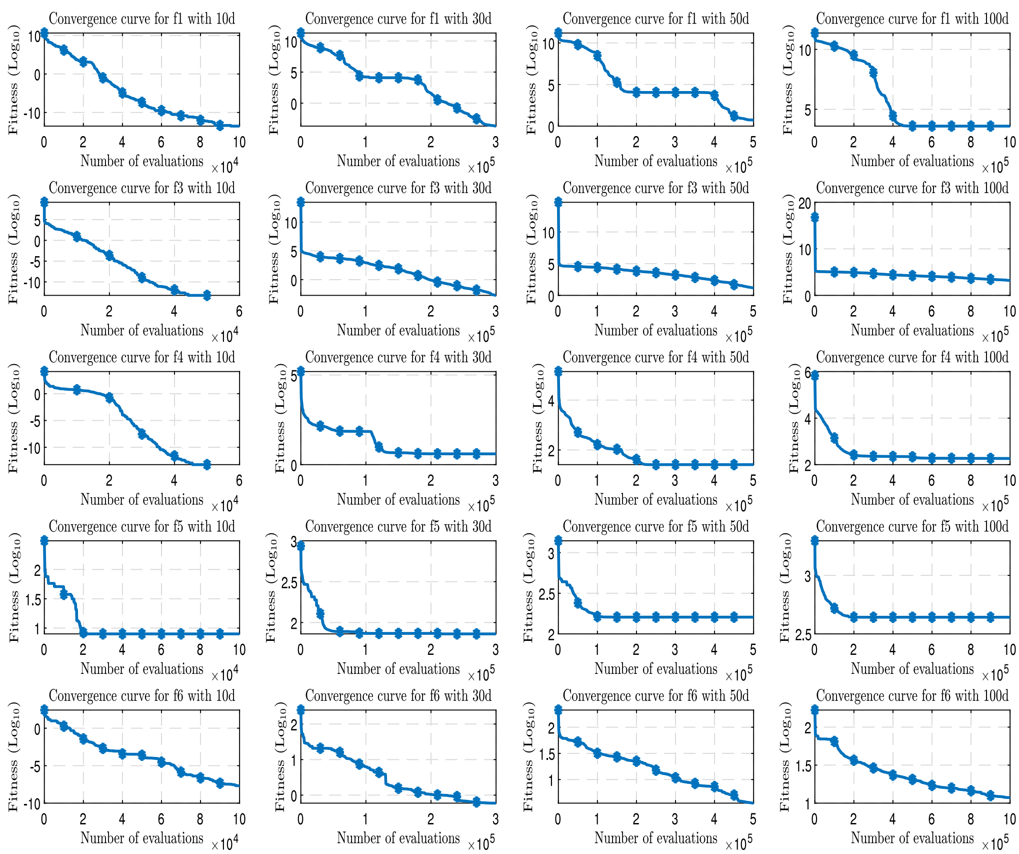
Holm's test method was then applied on the CEC-2017 test suite with 100d as a post-test method after Friedman's test. The statistical results of this test are presented in Table 27.

In Table 27, the hypotheses with a p-value $\leq$ 0.01666666 were rejected by Holm's test procedure. The results in this table show that WSO is one of the best optimization algorithms among i one when comparing this method with other ones.

In Table 27, the hypotheses with a p-value $\leq$ 0.01666666 were rejected by Holm's test procedure. The results in this table show that WSO is a promising optimization algorithm in solving high-dimensional problems, when it is compared to the most

Table 29
Results of Holm's test for the CEC-2011 test functions.

i	WSO vs.	z-value	p-value	α/i (0.05)	Hypothesis
9	ACO	6.17432701	6.64458872E−10	0.00555555	Rejected
8	PSO	5.27805373	1.30563181E−07	0.00625	Rejected
7	GA	3.90874734	9.27759293E−05	0.00714285	Rejected
6	DE	1.84233951	0.06542550	0.00833333	Not rejected
5	BBO	1.74275359	0.08137666	0.01	Not rejected
4	TLBO	0.77179087	0.44023829	0.0125	Not rejected
3	SFS	0.29875775	0.76512488	0.01666666	Not rejected
2	GSK	0.27386127	0.78419122	0.025	Not rejected
1	WSO	0.12448239	0.90093333	0.05	Not rejected



(a)

Fig. A.1. The incredible senses of a great white shark.

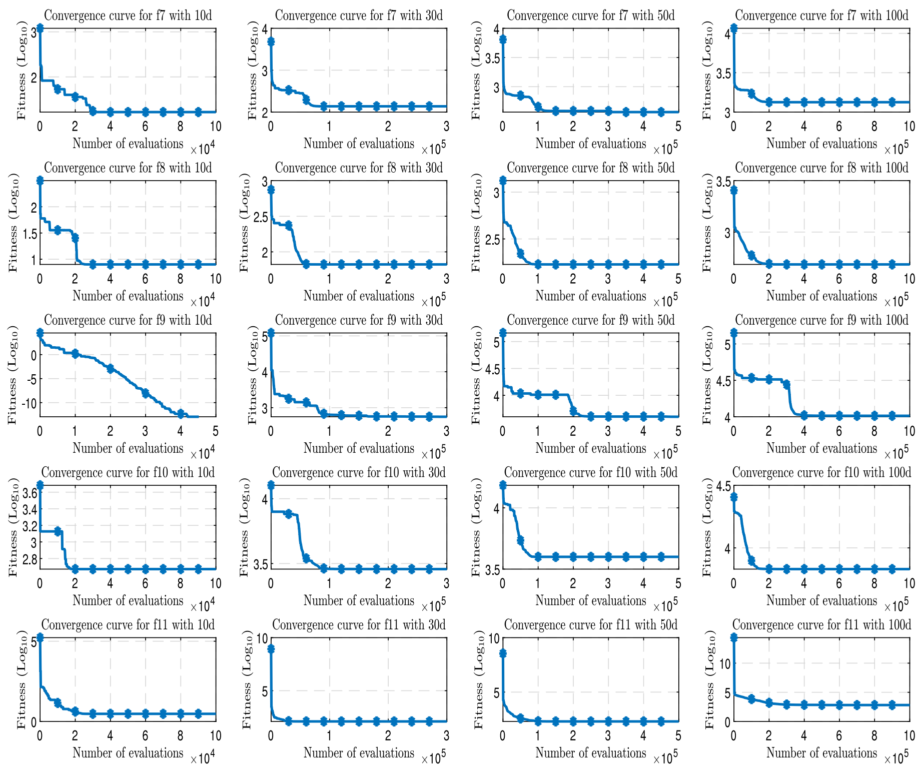
promising algorithms that scored the best performance in the literature.

Finally, Table 28 summarizes the ranking outcomes of WSO along with other competing methods by the application of Friedman's test with $\alpha = 0.05$ on the mean error results presented in Table 15.
The p-value computed by Friedman's test for the results of Table 28 is 6.55254739E−11. This substantiates that there is a statistically significant difference between the performances of the evaluated algorithms. According to the findings in Table 28, AMO is the control method that captured the best rank followed successively by WSO, GSK, SFS, TLBO, BBO, DE, GA, PSO, ACO. It is clear from Table 28 that WSO took the second rank among all algorithms, followed by GSK and SFS in third and fourth places, respectively. This implies that WSO is able to outpace promising optimization algorithms, where it reported a performance degree that significantly outperformed many other algorithms such as TLBO and DE.

To ensure that the differences between WSO and others are statistically large, Holm's test was applied, where its statistical results are displayed in Table 29.

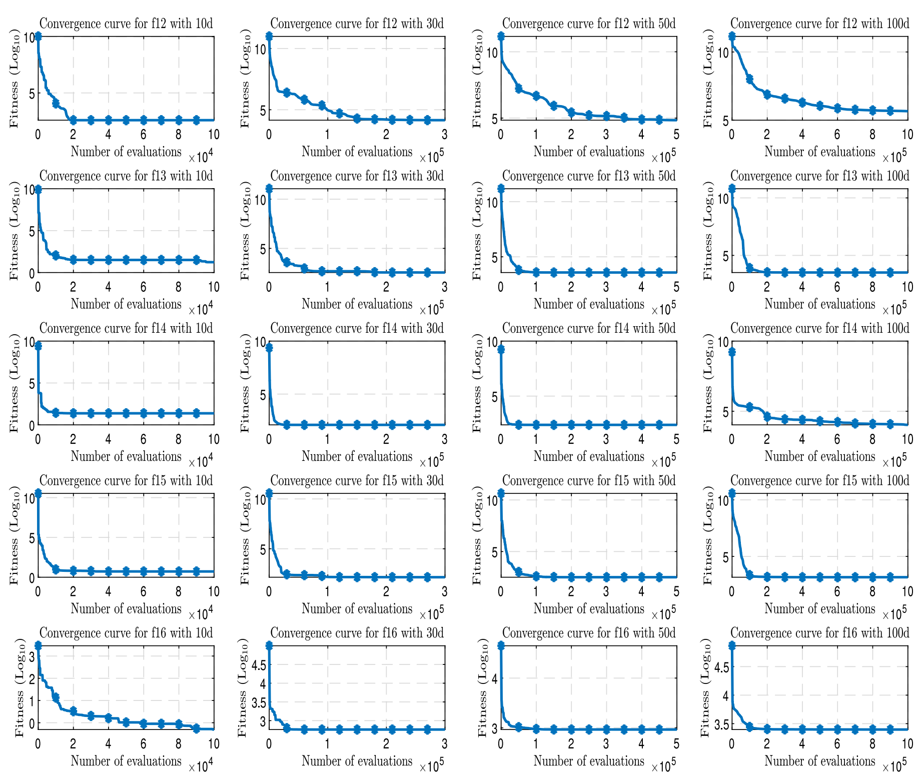
The hypotheses with p-value $\leq$ 0.00833333 in Table 29 are rejected based on Holm's method. It is observed from the findings of this test that WSO is able to provide a competitive performance like that reported by other promising algorithms mentioned in the literature.

As a key inference dragged from the statistical analysis results described above, in general, WSO performed better than many robust state-of-the-art methods aforesaid in the literature such as and TLBO, PSO and GA. This points out the favorable performance of WSO, and affirms that this algorithm can successfully explore the search space, whether it contains one optima or many optimums, or whether the optimization problems are of small, medium, or high dimensionality. Besides, the average ranking reveals that the performance score of WSO is not far behind that of AMO, GSK and SFS while the performance of WSO, AMO and GSK is far from all other competitors like PSO, GA and ACO.



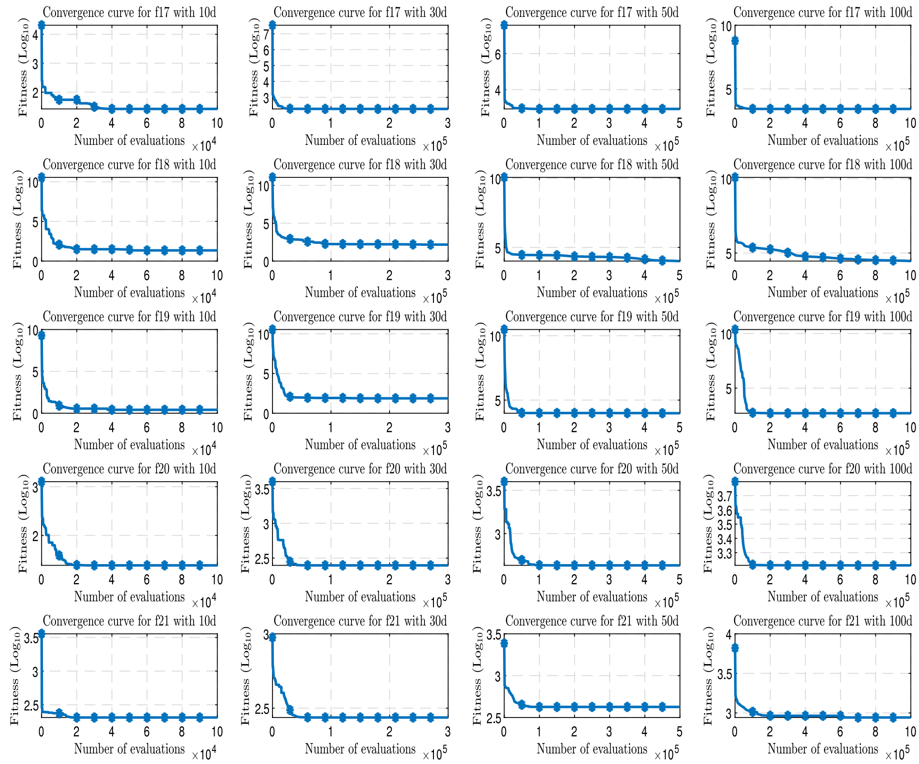
(b)

Fig. A.1. (*continued*).



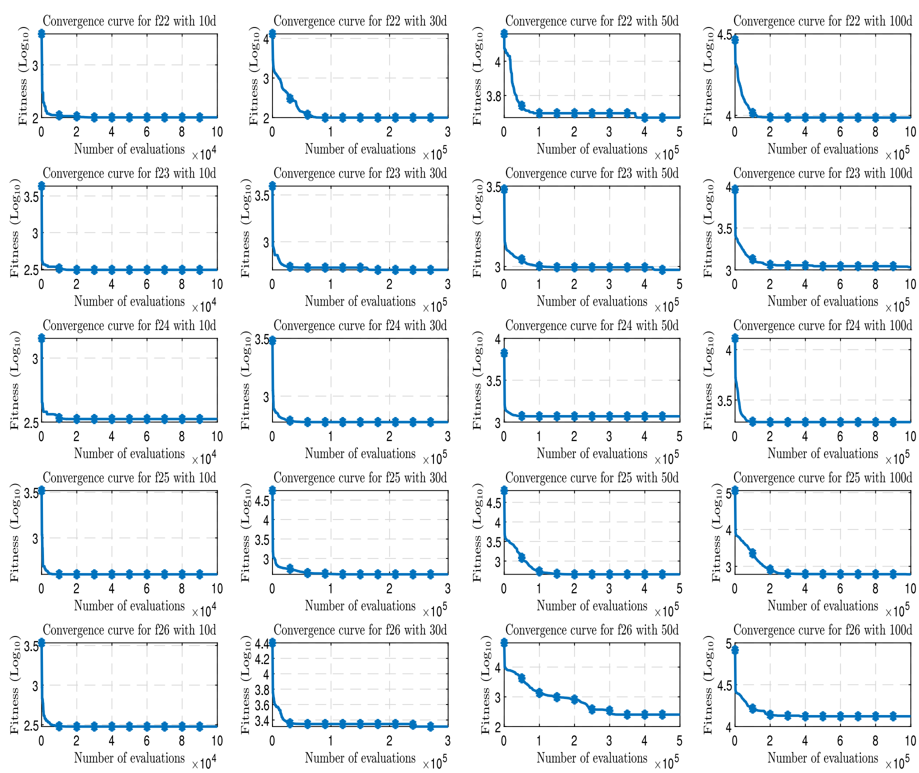
(c)

Fig. A.1. (*continued*).



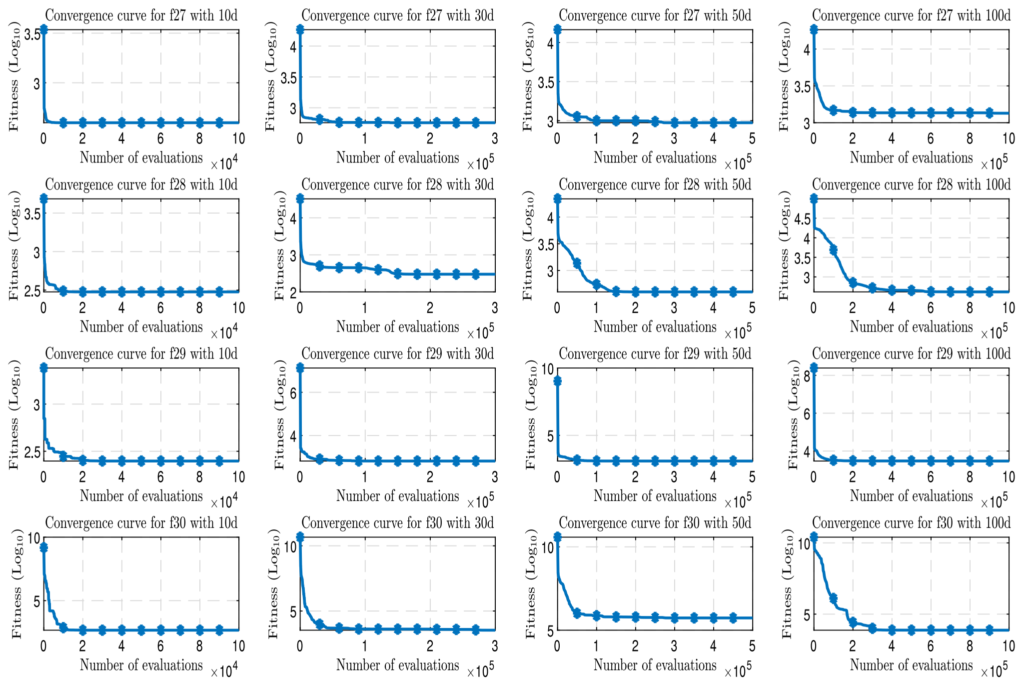
(d)

Fig. A.1. (*continued*).



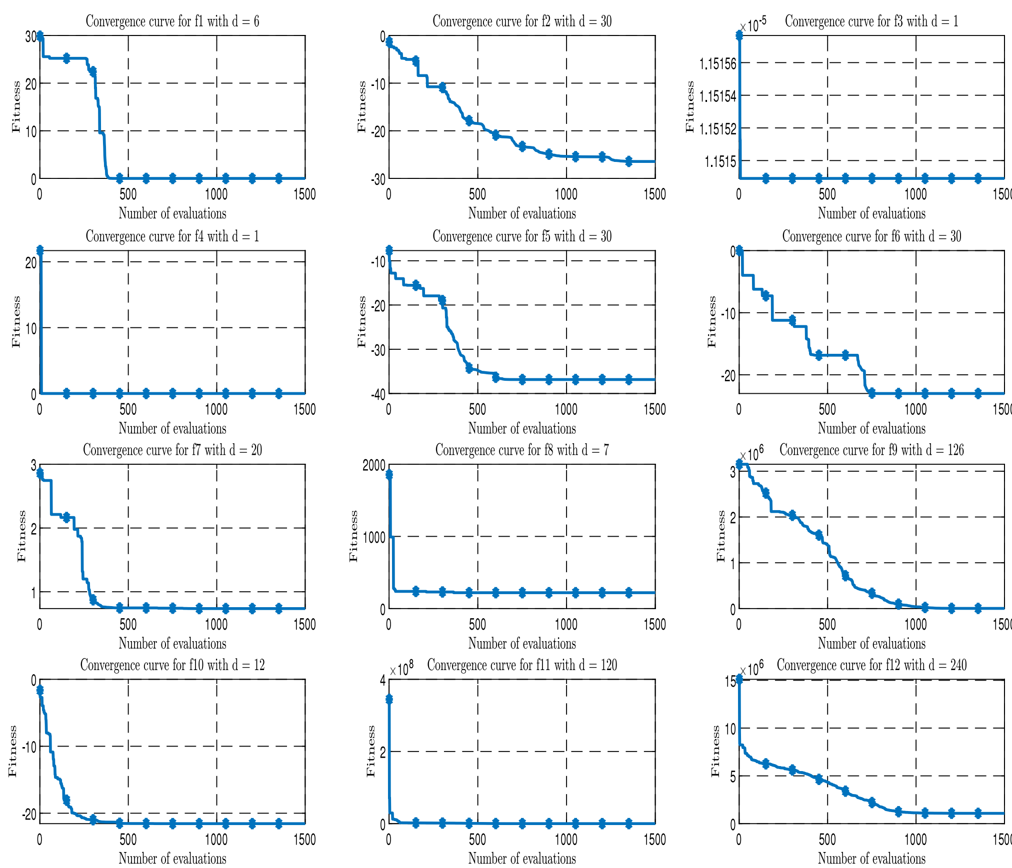
(e)

Fig. A.1. (*continued*).



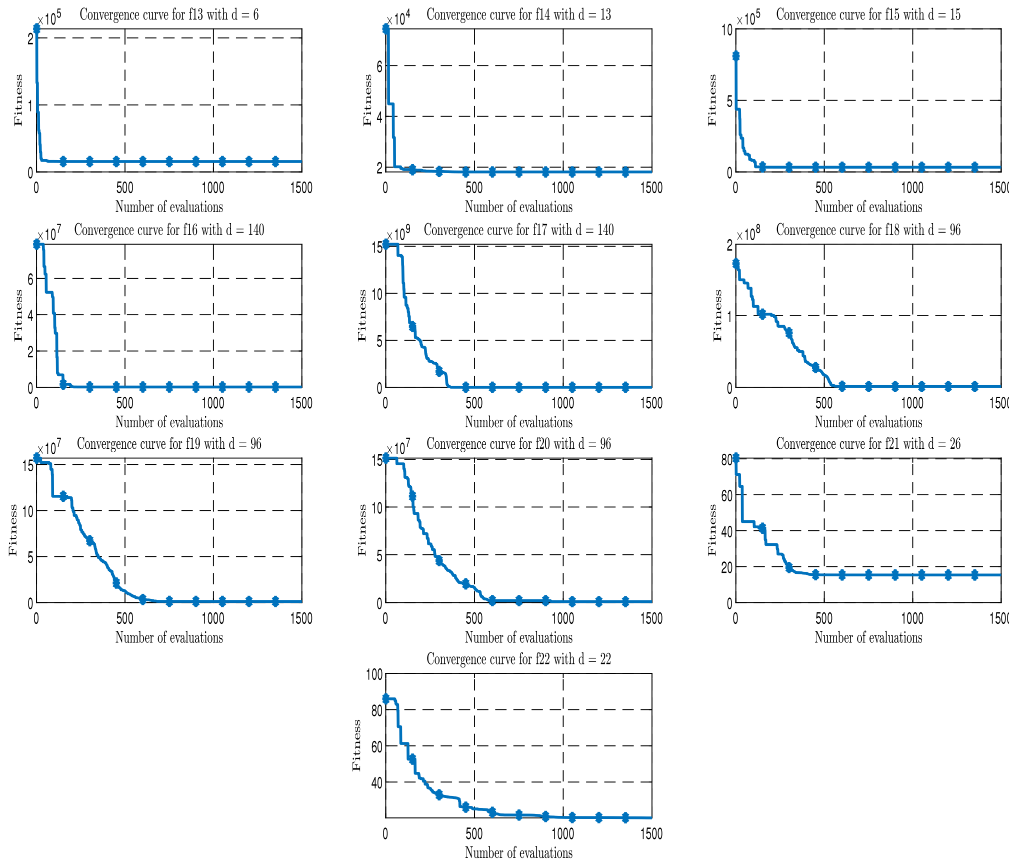
(f)

Fig. A.1. (*continued*).



(a)

Fig. A.2. Water flows through the lateral line systems, where vibrations in the water spur sensory cells in the main tube and alert the white shark to predators and prey.



(b)

Fig. A.2. (*continued*).

Precisely, we can draw that the outstanding superiority of WSO in CEC-2011 and CEC-2017 is attributable to the splendid and purposeful mathematical model of WSO. In sum, the consequences of this statistical analysis denote that WSO is an effective and reliable optimizer with well-tuned exploration and exploitation features that maintain a balance of local optimization and global optimization diversity. These endings are encouraging points to apply this algorithm to solve other hard real-world optimization problems.

5. Conclusion and future work

This paper has presented a novel meta-heuristic algorithm so-called White Shark Optimizer (WSO) to solve global optimization problems. The key concepts and inspiration of WSO stem from the behaviors of great white sharks, including their remarkable senses of hearing, sight and smell. The design of WSO incorporates promising exploratory and exploitative searches in its update mechanism to randomly update and change solutions. This behavior together with the appropriately defined locomotion force and smell strength terms of great white sharks were modeled to promote the exploratory conduct of WSO in the initial iterations and the exploitative search mechanism in the posterior iterations. The proposed WSO was assessed and experimented on a bunch of 29 benchmark test optimization problems with various dimensions belonging to the CEC-2017 benchmark test functions. Statistical test results on this test benchmark revealed the performance of WSO in reliably achieving the global optimal solutions with typically higher performance for most of the studied test problems compared to several well-studied optimization algorithms. WSO was further applied to solve the set of real-world optimization problems suggested for the CEC-2011 competition. In this real application, WSO performed considerably better than many other algorithms in the majority of test problems. Future works will adapt, carry out and test both multi-objective and binary versions of WSO to solve extreme high dimensions or large-scale real-world optimization problems. As WSO is effective in getting to the optimal or near-optimal point, a hybridization of WSO with other optimization methods may be considered as a potential additional search to further improve performance.

CRediT authorship contribution statement

Malik Braik: Proposed and evolved the mathematical models of the proposed algorithm, Prepared the experiments, tables, diagrams and pseudo-code of the proposed algorithm, Executed the programs and experimental scenarios of the work, Full revision of the entire paper. **Abdelaziz Hammouri:** Discussed all the computational results of the proposed algorithm and other compared algorithms, Conducted statistical tests and discussed their results. **Jaffar Atwan:** Discussed the convergence performance results of the proposed algorithm and other algorithms, Revised the abstract and conclusion of the paper. **Mohammed Azmi Al-Betar:** Checked the validation of the results and the references, Offered feedback on the work and helped shape and analyze the work. **Mohammed A. Awadallah:** Examined the technical concepts in the paper, the readability of the full paper and English grammar, Supported it in proof of English.

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.

Appendix. Convergence results of WSO

The Convergence graphs of the proposed WSO for the test functions of the CEC-2017 test suite with dimensions of 10, 30, 50 and 100 are presented in Fig. A.1. The Convergence graphs of the proposed WSO for the test problems of the CEC-2011 test suite are presented in Fig. A.2.

References

[1] Ali Wagdy Mohamed, Anas A Hadi, Ali Khater Mohamed, Gaining-sharing knowledge based algorithm for solving optimization problems: a novel nature-inspired algorithm, Int. J. Mach. Learn. Cybern. 11 (7) (2020) 1501–1529.

[2] Ehsan Jahani, Mohammad Chizari, Tackling global optimization problems with a novel algorithm–mouth brooding fish algorithm, Appl. Soft Comput. 62 (2018) 987–1002.

[3] Malik Braik, Alaa Sheta, Hamza Turabieh, Heba Alhiary, A novel lifetime scheme for enhancing the convergence performance of salp swarm algorithm, Soft Comput. (2020) 1–26.

[4] Malik Shehadeh Braik, Chameleon swarm algorithm: A bio-inspired optimizer for solving engineering design problems, Expert Syst. Appl. 174 (2021) 114685.

[5] Torben Ommen, Wiebke Brix Markussen, Brian Elmegaard, Comparison of linear, mixed integer and non-linear programming methods in energy system dispatch modelling, Energy 74 (2014) 109–118.

[6] Malik Braik, Alaa Sheta, Heba Al-Hiary, A novel meta-heuristic search algorithm for solving optimization problems: capuchin search algorithm, Neural Comput. Appl. (2020) 1–33.

[7] Laith Abualigah, Dalia Yousri, Mohamed Abd Elaziz, Ahmed A. Ewees, Mohammed A.A. Al-qaness, Amir H. Gandomi, Aquila optimizer: A novel meta-heuristic optimization algorithm, Comput. Ind. Eng. 157 (2021) 107250.

[8] E. Sandgren, Nonlinear Integer and Discrete Programming in Mechanical Design Optimization, 1990.

[9] Jiankai Xue, Bo Shen, A novel swarm intelligence optimization approach: sparrow search algorithm, Syst. Sci. Control Eng. 8 (1) (2020) 22–34.

[10] Seyedali Mirjalili, Amir H Gandomi, Seyedeh Zahra Mirjalili, Shahrzad Saremi, Hossam Faris, Seyed Mohammad Mirjalili, Salp swarm algorithm: A bio-inspired optimizer for engineering design problems, Adv. Eng. Softw. 114 (2017) 163–191.

[11] Mohit Jain, Vijander Singh, Asha Rani, A novel nature-inspired algorithm for optimization: Squirrel search algorithm, Swarm Evol. Comput. 44 (2019) 148–175.

[12] Cenker Aktemur, Islam Gusseinov, A comparison of sequential quadratic programming, genetic algorithm, simulated annealing, particle swarm optimization and hybrid algorithm for the design and optimization of golinski's speed reducer, Int. J. Energy Appl. Technol. 4 (2) (2017) 34–52.

[13] A Faramarzi, M.H. Afshar, Application of cellular automata to size and topology optimization of truss structures, Sci. Iran. 19 (3) (2012) 373–380.

[14] Malik Braik, Mohammad Hashem Ryalat, Hussein Al-Zoubi, A novel meta-heuristic algorithm for solving numerical optimization problems: Ali baba and the forty thieves, Neural Comput. Appl. (2021) 1–47.

[15] L.J. Fogel, aj Owens, m.j. walsh, Artificial Intelligence Through Simulated Evolution, 1966.

[16] Ingo Rechenberg, Evolution Strategy: Optimization of Technical Systems by Means of Biological Evolution, Vol. 104, Fromman-Holzboog, Stuttgart, 1973, pp. 15–16.

[17] John Henry Holland, et al., Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, MIT Press, 1992.

[18] Pablo Moscato, et al., On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms, Caltech Concurrent Computation Program, C3P Report, Vol. 826, 1989, p. 1989.

[19] W. Daniel Hillis, Co-evolving parasites improve simulated evolution as an optimization procedure, Physica D 42 (1–3) (1990) 228–234.

[20] John R Koza, John R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, Vol. 1, MIT Press, 1992.

[21] Robert G Reynolds, An introduction to cultural algorithms, in: Proceedings of the Third Annual Conference on Evolutionary Programming, Vol. 24, World Scientific, 1994, pp. 131–139.

[22] Rainer Storn, Kenneth Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341–359.

[23] Conor Ryan, John James Collins, Michael O. Neill, Grammatical evolution: Evolving programs for an arbitrary language, in: European Conference on Genetic Programming, Springer, 1998, pp. 83–96.

[24] Esmaeil Atashpaz-Gargari, Caro Lucas, Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition, in: 2007 IEEE Congress on Evolutionary Computation, Ieee, 2007, pp. 4661–4667.

[25] Pinar Civicioglu, Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm, Comput. Geosci. 46 (2012) 229–247.

[26] Pinar Civicioglu, Backtracking search optimization algorithm for numerical optimization problems, Appl. Math. Comput. 219 (15) (2013) 8121–8144.

[27] Hamid Salimi, Stochastic fractal search: a powerful metaheuristic algorithm, Knowl.-Based Syst. 75 (2015) 1–18.

[28] T.T Dhivyaprabha, P Subashini, M. Krishnaveni, Synergistic fibroblast optimization: a novel nature-inspired computing algorithm, Front. Inf. Technol. Electron. Eng. 19 (7) (2018) 815–833.

[29] Mohammad Mahdi Motevali, Ali Mohammadi Shanghooshabad, Reza Zohouri Aram, Hamidreza Keshavarz, Who: A new evolutionary algorithm bio-inspired by wildebeests with a case study on bank customer segmentation, Int. J. Pattern Recognit. Artif. Intell. 33 (05) (2019) 1959017.

[30] Chnoor M. Rahman, Tarik A. Rashid, A new evolutionary algorithm: Learner performance based behavior algorithm, Egypt. Inform. J. 22 (2) (2021) 213–223.

[31] Marco Dorigo, Vittorio Maniezzo, Alberto Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Trans. Syst. Man Cybern. B 26 (1) (1996) 29–41.

[32] James Kennedy, Russell Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-International Conference on Neural Networks, Vol. 4, IEEE, 1995, pp. 1942–1948.

[33] Hussein A. Abbass, Mbo: Marriage in honey bees optimization-a haplometrosis polygynous swarming approach, in: Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No. 01TH8546), Vol. 1, IEEE, 2001, pp. 207–214.

[34] Xiao-lei Li, An optimizing method based on autonomous animats: fish-swarm algorithm, Syst. Eng.-Theory Pract. 22 (11) (2002) 32–38.

[35] Kevin M. Passino, Biomimicry of bacterial foraging for distributed optimization and control, IEEE Control Syst. Mag. 22 (3) (2002) 52–67.

[36] Panta Lučić, Dušan Teodorović, Computing with bees: attacking complex transportation engineering problems, Int. J. Artif. Intell. Tools 12 (03) (2003) 375–394.

[37] Kaipa N. Krishnanand, Debasish Ghose, Detection of multiple source locations using a glowworm metaphor with applications to collective robotics, in: Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005, SIS 2005, IEEE, 2005, pp. 84–91.

[38] Shu-Chuan Chu, Pei-Wei Tsai, Jeng-Shyang Pan, Cat swarm optimization, in: Pacific Rim International Conference on Artificial Intelligence, Springer, 2006, pp. 854–858.

[39] Dervis Karaboga, Bahriye Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Global Optim. 39 (3) (2007) 459–471.

[40] Timothy C Havens, Christopher J Spain, Nathan G Salmon, James M. Keller, Roach infestation optimization, in: 2008 IEEE Swarm Intelligence Symposium, IEEE, 2008, pp. 1–7.

[41] Xin-She Yang, Suash Deb, Cuckoo search via lévy flights, in: Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on, IEEE, 2009, pp. 210–214.

[42] Shan He, Q. Henry Wu, J.R. Saunders, Group search optimizer: an optimization algorithm inspired by animal searching behavior, IEEE Trans. Evol. Comput. 13 (5) (2009) 973–990.

[43] Xin-She Yang, A new metaheuristic bat-inspired algorithm, in: Nature Inspired Cooperative Strategies for Optimization, NICSO 2010, Springer, 2010, pp. 65–74.

[44] Xin-She Yang, Firefly algorithms for multimodal optimization, in: International Symposium on Stochastic Algorithms, Springer, 2009, pp. 169–178.

[45] Ramin Hedayatzadeh, Foad Akhavan Salmassi, Manijeh Keshtgari, Reza Akbari, Koorush Ziarati, Termite colony optimization: A novel approach for optimizing continuous problems, in: 2010 18th Iranian Conference on Electrical Engineering, IEEE, 2010, pp. 553–558.

[46] Wen-Tsao Pan, A new fruit fly optimization algorithm: taking the financial distress model as an example, Knowl.-Based Syst. 26 (2012) 69–74.

[47] Xin-She Yang, Flower pollination algorithm for global optimization, in: International Conference on Unconventional Computing and Natural Computation, Springer, 2012, pp. 240–249.

[48] Amir Hossein Gandomi, Amir Hossein Alavi, Krill herd: a new bio-inspired optimization algorithm, in: Communications in Nonlinear Science and Numerical Simulation, Vol. 17, (12) 2012, pp. 4831–4845.

[49] A Kaveh, N. Farhoudi, A new optimization method: dolphin echolocation, Adv. Eng. Softw. 59 (2013) 53–70.

[50] Mehdi Neshat, Ghodrat Sepidnam, Mehdi Sargolzaei, Swallow swarm optimization algorithm: a new method to optimization, Neural Comput. Appl. 23 (2) (2013) 429–454.
[51] Xiangtao Li, Jie Zhang, Minghao Yin, Animal migration optimization: an optimization algorithm inspired by animal migration behavior, Neural Comput. Appl. 24 (7) (2014) 1867–1877.
[52] Seyedali Mirjalili, The ant lion optimizer, Adv. Eng. Softw. 83 (2015) 80–98.
[53] Sait Ali Uymaz, Gulay Tezel, Esra Yel, Artificial algae algorithm (aaa) for nonlinear global optimization, Appl. Soft Comput. 31 (2015) 153–171.
[54] Seyedali Mirjalili, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, Neural Comput. Appl. 27 (4) (2016) 1053–1073.
[55] Mu Dong Li, Hui Zhao, Xing Wei Weng, Tong Han, A novel nature-inspired algorithm for optimization: Virus colony search, Adv. Eng. Softw. 92 (2016) 65–88.
[56] Alireza Askarzadeh, A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm, Comput. Struct. 169 (2016) 1–12.
[57] Wang Yong, Wang Tao, Zhang Cheng-Zhi, Huang Hua-Juan, A new stochastic optimization approach–dolphin swarm optimization algorithm, Int. J. Comput. Intell. Appl. 15 (02) (2016) 1650011.
[58] Seyedali Mirjalili, Andrew Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.
[59] Shahrzad Saremi, Seyedali Mirjalili, Andrew Lewis, Grasshopper optimisation algorithm: theory and application, Adv. Eng. Softw. 105 (2017) 30–47.
[60] Gaurav Dhiman, Vijay Kumar, Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications, Adv. Eng. Softw. 114 (2017) 48–70.
[61] Ali Asghar Heidari, Seyedali Mirjalili, Hossam Faris, Ibrahim Aljarah, Majdi Mafarja, Huiling Chen, Harris hawks optimization: Algorithm and applications, Future Gener. Comput. Syst. 97 (2019) 849–872.
[62] Tapabrata Ray, Kim-Meow Liew, Society and civilization: an optimization algorithm based on the simulation of social behavior, IEEE Trans. Evol. Comput. 7 (4) (2003) 386–396.
[63] W. Chen, C. Dai, Y. Zhu, Seeker optimization algorithm, in: International Conference on Computational and Information Science, Springer, 2006, pp. 167–176.
[64] C. Lucas, E. Atashpaz-Gargari, Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition, in: 2007 IEEE Congress on Evolutionary Computation, IEEE, 2007, pp. 4661–4667.
[65] Luna Mingyi Zhang, Cheyenne Dahlmann, Yanqing Zhang, Human-inspired algorithms for continuous function optimization, in: 2009 IEEE International Conference on Intelligent Computing and Intelligent Systems, Vol. 1, IEEE, 2009, pp. 318–321.
[66] Ali Husseinzadeh Kashan, League championship algorithm: a new algorithm for numerical function optimization, in: 2009 International Conference of Soft Computing and Pattern Recognition, IEEE, 2009, pp. 43–48.
[67] Yuechun Xu, Zhihua Cui, Jianchao Zeng, Social emotional optimization algorithm for nonlinear constrained optimization problems, in: International Conference on Swarm, Evolutionary, and Memetic Computing, Springer, 2010, pp. 583–590.
[68] Yuhui Shi, Brain storm optimization algorithm, in: International Conference in Swarm Intelligence, Springer, 2011, pp. 303–309.
[69] R. Venkata Rao, Vimal J Savsani, D.P. Vakharia, Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems, Inform. Sci. 183 (1) (2012) 1–15.
[70] H Shayeghi, J. Dadashpour, Anarchic society optimization based pid control of an automatic voltage regulator (avr) system, Electr. Electron. Eng. 2 (4) (2012) 199–207.
[71] Seyed Jalaleddin Mousavirad, Hossein Ebrahimpour-Komleh, Human mental search: a new population-based metaheuristic optimization algorithm, Appl. Intell. 47 (3) (2017) 850–887.
[72] Reza Moghdani, Khodakaram Salimifard, Volleyball premier league algorithm, Appl. Soft Comput. 64 (2018) 161–185.
[73] Mohammed Azmi Al-Betar, Zaid Abdi Alkareem Alyasseri, Mohammed A Awadallah, Iyad Abu Doush, Coronavirus herd immunity optimizer (chio), Neural Comput. Appl. 33 (10) (2021) 5011–5042.
[74] Afshin Faramarzi, Mohammad Heidarinejad, Brent Stephens, Seyedali Mirjalili, Equilibrium optimizer: A novel optimization algorithm, Knowl.-Based Syst. 191 (2020) 105190.
[75] Scott Kirkpatrick, C. Daniel Gelatt, Mario P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.
[76] Nenad Mladenović, Pierre Hansen, Variable neighborhood search, Comput. Oper. Res. 24 (11) (1997) 1097–1100.
[77] Osman K Erol, Ibrahim Eksin, A new optimization method: big bang–big crunch, Adv. Eng. Softw. 37 (2) (2006) 106–111.
[78] Richard A. Formato, Central force optimization, Prog. Electromagn. Res. 77 (1) (2007) 425–491.
[79] Hamed Shah Hosseini, Problem solving by intelligent water drops, in: 2007 IEEE Congress on Evolutionary Computation, IEEE, 2007, pp. 3226–3231.
[80] David R. Monismith, Blayne E. Mayfield, Slime mold as a model for numerical optimization, in: 2008 IEEE Swarm Intelligence Symposium, IEEE, 2008, pp. 1–8.
[81] Esmat Rashedi, Hossein Nezamabadi-Pour, Saeid Saryazdi, Gsa: a gravitational search algorithm, Inform. Sci. 179 (13) (2009) 2232–2248.
[82] A Kaveh, Siamak Talatahari, A novel heuristic optimization method: charged system search, Acta Mech. 213 (3) (2010) 267–289.
[83] Erik Cuevas, Diego Oliva, Daniel Zaldivar, Marco Pérez-Cisneros, Humberto Sossa, Circle detection using electro-magnetism optimization, Inform. Sci. 182 (1) (2012) 40–55.
[84] Abdolreza Hatamlou, Black hole: A new heuristic optimization approach for data clustering, Inform. Sci. 222 (2013) 175–184.
[85] Hadi Eskandar, Ali Sadollah, Ardeshir Bahreininejad, Mohd Hamdi, Water cycle algorithm–a novel metaheuristic optimization method for solving constrained engineering optimization problems, Comput. Struct. 110 (2012) 151–166.
[86] Ali Sadollah, Ardeshir Bahreininejad, Hadi Eskandar, Mohd Hamdi, Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems, Appl. Soft Comput. 13 (5) (2013) 2592–2612.
[87] Ali Kaveh, Vahid Reza Mahdavi, Colliding bodies optimization: a novel meta-heuristic method, Comput. Struct. 139 (2014) 18–27.
[88] Shareef Hussain, A.A. Ibrahim, A.H. Mutlag, Lightning search algorithm, Appl. Soft Comput. 36 (2015) 315–333.
[89] Adil Baykasoğlu, Şener Akpinar, Weighted superposition attraction (wsa): A swarm intelligence algorithm for optimization problems–part 1: Unconstrained optimization, Appl. Soft Comput. 56 (2017) 520–540.
[90] Seyedali Mirjalili, Seyed Mohammad Mirjalili, Abdolreza Hatamlou, Multi-verse optimizer: a nature-inspired algorithm for global optimization, Neural Comput. Appl. 27 (2) (2016) 495–513.
[91] A Kaveh, Armin Dadras, A novel meta-heuristic optimization algorithm: thermal exchange optimization, Adv. Eng. Softw. 110 (2017) 69–84.
[92] Ali Husseinzadeh Kashan, Reza Tavakkoli-Moghaddam, Mitsuo Gen, Find-fix-finish-exploit-analyze (f3ea) meta-heuristic algorithm: An effective algorithm with new evolutionary operators for global optimization, Comput. Ind. Eng. 128 (2019) 192–218.
[93] Fatma A Hashim, Essam H Houssein, Mai S Mabrouk, Walid Al-Atabany, Seyedali Mirjalili, Henry gas solubility optimization: A novel physics-based algorithm, Future Gener. Comput. Syst. 101 (2019) 646–667.
[94] Fatma A Hashim, Kashif Hussain, Essam H Houssein, Mai S Mabrouk, Walid. Al-Atabany, Archimedes optimization algorithm: a new meta-heuristic algorithm for solving optimization problems, Appl. Intell. 51 (3) (2021) 1531–1551.
[95] João Luiz Junho Pereira, Matheus Brendon Francisco, Camila Aparecida Diniz, Guilherme Antônio Oliver, Jr Sebastiao Simoes Cunha, Guilherme Ferreira Gomes, Lichtenberg algorithm: A novel hybrid physics-based meta-heuristic for global optimization, Expert Syst. Appl. 170 (2021) 114522.
[96] Bernardo Morales-Castañeda, Daniel Zaldivar, Erik Cuevas, Fernando Fausto, Alma Rodríguez, A better balance in metaheuristic algorithms: Does it exist? Swarm Evol. Comput. 54 (2020) 100671.
[97] Malik Braik, Alaa Sheta, Exploration of genetic algorithms and particle swarm optimization in improving the quality of medical images, 2011.
[98] Dexuan Zou, Steven Li, Xiangyong Kong, Haibin Ouyang, Zongyan Li, Solving the dynamic economic dispatch by a memory-based global differential evolution and a repair technique of constraint handling, Energy 147 (2018) 59–80.
[99] Alaa Sheta, Malik Braik, Heba Al-Hiary, Modeling the tennessee eastman chemical process reactor using bio-inspired feedforward neural network (bi-ff-nn), Int. J. Adv. Manuf. Technol. 103 (1–4) (2019) 1359–1380.
[100] Ahmed Mahdi Jubair, Rosilah Hassan, Azana Hafizah Mohd Aman, Hasimi Sallehudin, Social class particle swarm optimization for variable-length wireless sensor network deployment, Appl. Soft Comput. 113 (2021) 107926.
[101] Dexuan Zou, Dunwei. Gong, Differential evolution based on migrating variables for the combined heat and power dynamic economic dispatch, Energy 238 (2022) 121664.
[102] Manik Sharma, Prableen Kaur, A comprehensive analysis of nature-inspired meta-heuristic techniques for feature selection problem, Arch. Comput. Methods Eng. 28 (3) (2021).
[103] David H Wolpert, William G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82.
[104] Stephen Wroe, D.R Huber, M Lowry, Colin McHenry, Karen Moreno, Philip Clausen, Toni L Ferrara, Eleanor Cunningham, M.N Dean, A.P. Summers, Three-dimensional computer analysis of white shark jaw mechanics: how hard can a great white bite? J. Zool. 276 (4) (2008) 336–342.

[105] Demian D Chapman, Debra L Abercrombie, Christophe J Douady, Ellen K Pikitch, Michael J Stanhopen, Mahmood S. Shivji, A streamlined, bi-organelle, multiplex pcr approach to species identification: Application to global conservation and trade monitoring of the great white shark, carcharodon carcharias, Conserv. Genet. 4 (4) (2003) 415–425.
[106] Noor H Awad, Mostafa Z Ali, Ponnuthurai 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, pp. 372–379.
[107] Swagatam Das, Ponnuthurai N. Suganthan, Problem Definitions and Evaluation Criteria for Cec 2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Problems, Jadavpur University, Nanyang Technological University, Kolkata, 2010, pp. 341–359.
[108] Eric Bonabeau, Marco Dorigo, Directeur de Recherches Du Fnrs Marco, Guy Theraulaz, Guy Théraulaz, et al., Swarm Intelligence: From Natural to Artificial Systems, No. 1, Oxford University Press, 1999.
[109] Dan. Simon, Biogeography-based optimization, IEEE Trans. Evol. Comput. 12 (6) (2008) 702–713.
[110] James Kennedy, Russell Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-International Conference on Neural Networks, Vol. 4, IEEE, 1995, pp. 1942–1948.
[111] Ali W Mohamed, Anas A Hadi, Anas M Fattouh, Kamal M. Jambi, Lshade with semi-parameter adaptation hybrid with cma-es for solving cec 2017 benchmark problems, in: 2017 IEEE Congress on Evolutionary Computation, CEC, IEEE, 2017, pp. 145–152.
[112] Dulce G Pereira, Anabela Afonso, Fátima Melo Medeiros, Overview of friedman's test and post-hoc analysis, Comm. Statist. Simulation Comput. 44 (10) (2015) 2636–2653.
[113] Janez Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (Jan) (2006) 1–30.
[114] Sture Holm, A simple sequentially rejective multiple test procedure, Scand. J. Stat. (1979) 65–70.