



Available online at [www.sciencedirect.com](http://www.sciencedirect.com)

**ScienceDirect**

Comput. Methods Appl. Mech. Engrg. 403 (2023) 115652

**Computer methods  
in applied  
mechanics and  
engineering**

[www.elsevier.com/locate/cma](http://www.elsevier.com/locate/cma)

# Young's double-slit experiment optimizer : A novel metaheuristic optimization algorithm for global and constraint optimization problems

Mohamed Abdel-Basset<sup>a</sup>, Doaa El-Shahat<sup>a</sup>, Mohammed Jameel<sup>b,c</sup>,  
Mohamed Abouhawwash<sup>b,d,\*</sup>

<sup>a</sup> Faculty of Computers and Informatics, Zagazig University, Shaibet an Nakareyah, Zagazig, 44519 Ash Sharqia Governorate, Egypt

<sup>b</sup> Department of Mathematics, Faculty of Science, Mansoura University, Mansoura 35516, Egypt

<sup>c</sup> Department of Mathematics, Faculty of Science, Sana'a University, Sana'a 13509, Yemen

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

Received 31 July 2022; received in revised form 2 September 2022; accepted 12 September 2022

Available online 4 November 2022

Dataset link: [https://www.researchgate.net/publication/364243097\\_YDSE\\_Codes](https://www.researchgate.net/publication/364243097_YDSE_Codes)

## Abstract

Due to the global progress, the optimization problems are becoming more and more complex. Hence, deterministic and heuristic approaches are no longer adequate for dealing with such sophisticated problems. subsequently, metaheuristics have recently emerged as an effective alternative for addressing the optimization problems. This paper proposes a novel metaheuristic called Young's Double-Slit Experiment (YDSE) optimizer, derived from a physical backdrop. The YDSE optimizer is inspired by Young's double-slit experiment, which is regarded as one of the most well-known classical physics experiments, revealing the wave nature of light. In YDSE optimizer, each fringe represents a possible solution in the population. Many concepts are modeled from the experiment, such as monochromatic light waves, Huygens' principle, constructive and destructive interference, wave intensity, amplitude, and path difference. The YDSE optimizer strikes a balance between exploration and exploitation by selecting either a constructive interference or a destructive interference based on the order number of the fringe. During the optimization process, the solution moves in search space based on its order number. If the solution has an odd number, it moves in the dark regions towards the central bright region, which is expected to contain the optimal solution. The algorithm exploits the promising areas in the bright fringe areas, which are assumed to contain the optimum. The performance of the YDSE optimizer is compared with another twelve metaheuristics using CEC 2014, CEC 2017, and CEC 2022. The benchmarks cover different unimodal, multimodal, hybrid, and composite test functions. Also, we consider ten constrained and unconstrained engineering optimization design problems. YDSE proved its superiority over the CEC 2014 and CEC 2017 winners, such as L-SHADE, LSHADE-cnEpSin, and LSHADE-SPACMA. The results and the statistical analysis demonstrated the outperformance of the proposed YDSE optimizer at a 95% confidence interval.

Published by Elsevier B.V.

**Keywords:** Young's Double-Slit Experiment; Huygens' principle; Light wave; Interference; Metaheuristics; Constrained optimization

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

E-mail addresses: [mohamedbasset@ieee.org](mailto:mohamedbasset@ieee.org) (M. Abdel-Basset), [doaaelshahat@zu.edu.eg](mailto:doaaelshahat@zu.edu.eg) (D. El-Shahat), [moh.jameel@su.edu.ye](mailto:moh.jameel@su.edu.ye) (M. Jameel), [abouhaww@msu.edu](mailto:abouhaww@msu.edu), [saleh1284@mans.edu.eg](mailto:saleh1284@mans.edu.eg) (M. Abouhawwash).

## 1. Introduction

Numerous industries and sectors, such as manufacturing [1], agriculture [2], tourism [3], and energy [4], have comprised various optimization issues. These optimization issues face nonlinearity, discontinuity, uncertainty, high dimensionality, multi-objective, and non-convex challenges. Furthermore, human and industrial growth posed considerable hurdles to the current optimization methods to solve such up-to-date optimization problems. Hence, numerous optimization approaches have been proposed: deterministic, heuristic, and metaheuristic algorithms.

The deterministic algorithms [5] primarily employ particular mathematical rules, like gradient descent [6], Newton and quasi-Newton techniques [7], Nelder-and-Mead's simplex approach [8], and Levenberg–Marquardt [9]. These types of methods always return the same solution for the specific input to a given issue, and they need some gradient and Hessian information about the optimization issues. They can find the local optimum when confronted with highly constrained and multimodal problems. However, compared to meta-heuristic approaches, these methods are more rapid in identifying a solution.

Unlike the deterministic methods, the heuristic and metaheuristic methods have randomness values in their structure. Therefore, the heuristic methods did not guarantee the optimal solutions, but they obtained quick and near-optimal solutions, in contrast to the exact methods. Even though the exact methods [10,11] provide the optimal results, they take a long time to solve large-scale issues. One disadvantage of heuristic approaches is that these methods' effectiveness is firmly contingent on the initial starting point.

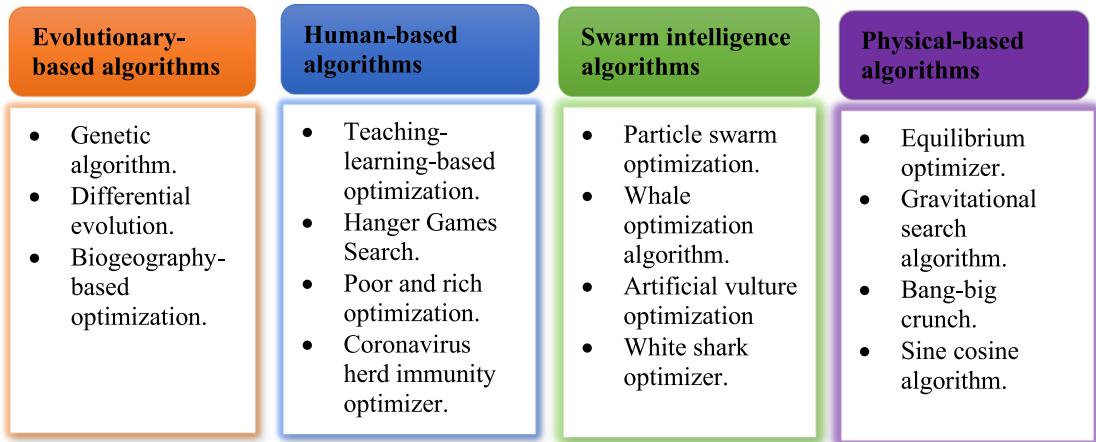
Metaheuristics (MH) are another category of algorithms that mimic the natural behaviors of beings or phenomena by primarily using randomness and mathematical operators. In recent years, several metaheuristics have evolved, drawing the interest of many scholars and proven to be effective in tackling many optimization issues: feature selection [12,13], wireless sensor network [14], knapsack problem [15], traveling salesman problem [16], intrusion detection system [17] and task scheduling [18]. Due to the inadequacies of previously stated deterministic and heuristic methods, MHs have emerged as a perfect alternative for addressing optimization problems. Most studies classified MHs into four categories [19,20]: Evolutionary-based Algorithms (EAs), swarm intelligence (SI) algorithms, Human-based Algorithms (HAs), and physics-based algorithms (PA), as shown in Fig. 1.

## 2. Related work

In this section, we will review the four categories of MHs. EAs are population-based algorithms built using the following genetic operators: mutation, selection, crossover, and elimination. One of the earliest EAs to be created was the Genetic Algorithm (GA). Generally, the crossover rate, mutation rate, the objective function, and population size commonly indicate the GA performance [21]. Differential Evolution (DE) [22] is another evolutionary algorithm. Lately, Nadimi-Shahraki et al. [23] integrated the DE with the concept of a multi-trial vector to combine different search strategies on the subpopulation using a distribution policy. In this regard, fast evolutionary programming [24,25], estimation of distribution algorithms [26], biogeography-based optimization [27], and memetic algorithms [28] are among other EAs. Furthermore, Tang et al. [29] developed a novel EA that imitates the invasive tumor growth behavior such that each tumor cell strives for nutrients in its microenvironment to grow and reproduce.

SI algorithms are multi-agent systems based on the cooperative and intelligent interactions between the living groups of insects or animals. In this field, Particle Swarm Optimization (PSO) [30] stands out as one of the first SI algorithms, which mimics the collective behavior of particles searching for nourishment. Recently, the researchers improved the performance of PSO using technologies such as three archives [31], Lévy flight [32], and mutation [33]. The artificial bee colony is another SI algorithm designed by Karaboga and Basturk [34] to simulate the intelligent foraging behaviors of bees with the collaboration of Scout bees, employed bees, and onlooker bees. Unfortunately, this algorithm had flaws like poor exploitation and slow convergence [35–37].

Furthermore, the cuckoo search [38] mimics obligatory brood parasite behavior in some cuckoo species. Also, Dhiman et al. [39] created a **Rat Swarm Optimizer** (RSO) based on the pursuing and attacking behaviors of rats in nature. Another SI algorithm is the White Shark Optimizer (WSO) [40], which relies on the complex traits of the great white sharks. Based on African vultures' feeding and navigational habits, the **Artificial Vulture Optimization Algorithm** (AVOA) [41] was developed. Humpback whales serve as inspiration for a new algorithm called the Whale Optimization Algorithm (WOA) [42]. Another algorithm [43] is developed based on the behavior of the Beluga whale. A SI algorithm inspired by starling actions during their astonishing murmuration is introduced in [20]. In



**Fig. 1.** Classification of MH algorithms.

addition, more SI algorithms were developed, including marine predators algorithm [44], grey wolf optimizer [45], firefly algorithm [46], moth flame optimization [47], salp swarm algorithm [48], golden eagle optimizer [49], and squirrel search algorithm [50]. Some algorithms may be hybridized with each other to exploit their strength points [51,52].

Several MH algorithms were designed by taking inspiration from the physical laws behind distinct phenomena. Rashedi et al. [53] proposed the gravitational search algorithm relying on the interaction among a number of masses using the laws of gravity and motion. Also, Faramarzi et al. [54] used the theories of control volume mass balance to propose an Equilibrium Optimizer (EO). The Slime Mould Algorithm (SMA) [55] was conceived based on the oscillation mode of slime mould seen in nature. Runge Kutta method (RUN) [56] used the slope variations calculated using the Runge Kutta method in mathematics. Mirjalili [57] suggested a MH algorithm using the sine and cosine functions. Moreover, Abualigah et al. [58] established an algorithm with the help of the arithmetic operations (+, -, \*, /). There were many well-regarded examples of PA stated as follows: big bang-big crunch [59], Lichtenberg algorithm [60], multi-verso optimization [61], and generalized normal distribution optimization algorithm [62].

Human beings are the most intelligent creatures. Hence, various MH algorithms were inspired by human social behaviors and activities, such as sports, social behaviors, politics, and diseases. For example, and not as a limitation, Table 1 presents some of these algorithms with details like type, abbreviation, and inspiration.

Despite a plethora of nature-inspired optimizers, we may wonder why new algorithms are constantly being created. From the No Free Lunch (NFL) theorem [80], there is not an optimization algorithm that can efficiently tackle all the optimization problems with the same capacity. As a result, there will be several attempts by researchers to design more effective and robust MHs for tackling optimization problems. This motivates us to suggest a new optimization algorithm inspired by a well-known physical experiment. Our main contributions to the paper can be summarized as follows:

- The YDSE optimizer is a novel MH inspired by the physical Young's double-slit experiment that demonstrates the wave nature of the light. Firstly, a random population is generated to represent the monochromatic light waves. When these waves pass through the two slits, they spread out in all directions from each slit according to the Huygens' principle.
- The interference patterns are constructed based on points created on the two wavefronts outgoing from the two slits. In YDSE, the constructive and destructive interference points result in bright and dark fringes on the projection screen. Each fringe represents a possible solution in the final population.
- During the optimization process, the solution moves in the search space based on its order number. If it has an odd number, it moves in the dark regions towards the bright central region, which is expected to contain the optimal solution. In the exploitation phase, the algorithm exploits the promising areas in the bright fringe areas, which are assumed to contain the optimum. For the exploration phase, if the solution has an odd number,

**Table 1**

Summarization of many human-based algorithms.

Type	Algorithm	Abbrev.	Ref.	Inspiration
Social behaviors	Teaching-learning-based optimization	TLBO	[63]	The teacher and the learners
	Social-based algorithm	SBA	[64]	People in different communities
	Poor and rich optimization	PRO	[65]	Poor and rich people to get wealth
	Social group optimization	SGO	[66]	Human behaviors to face complicated problems
	Nomadic people optimizer	NPO	[67]	Nomadic people conduct
Sport	Soccer league competition	SLC	[68]	Soccer leagues competitions
	League championship algorithm	LGA	[69]	Sport champions
	World cup optimization algorithm	WCO	[70]	FIFA world cup football competition
	Ludo game-based swarm intelligence	LGSI	[71]	The rules of playing Ludo game
	Football game algorithm	FGA	[72]	Behaviors of football players to score a goal
Politics	Election campaign algorithm	ECA	[73]	Candidates conduct to get support in the campaign
	Political optimizer	PO	[74]	Multi-phased process of politics
	Greedy politics optimization	GPO	[75]	Elections for government formation
	Imperialist competitive algorithm	ICA	[76]	Imperialistic competition
	Parliamentary political competition	PPC	[77]	Political competitions in parliamentary elections
Farming	Farmland fertility algorithm	FFA	[78]	Division of farmland based on their fertility
Disease	Coronavirus herd immunity optimizer	CHIO	[79]	Herd immunity to tackle COVID-19 virus

it advances through the dark regions towards the central zone, which is thought to have the best solution to skip from the local optima.

- The performance of the YDSE optimizer is tested against another twelve MHs using CEC 2014, CEC 2017, CEC 2022, and ten constrained and unconstrained engineering design problems. YDSE outperformed the majority of algorithms in addition to the CEC 2014 and CEC 2017 winners, such as L-SHADE, LSHADE-cnEpSin, and LSHADE-SPACMA. The results and the statistical analysis demonstrated the superiority of the proposed YDSE optimizer at a 95% confidence interval.

The paper sections are organized as follows. In Section 2, we present the previous works of MHs. Section 3 discusses the experiment of Young's double-slit as a primary inspiration for the proposed algorithm. Moreover, the proposed algorithm is illustrated in Section 4. Section 5 provides the results and statistical analyses of many algorithms compared to the proposed algorithm. The final sixth section summarizes some conclusions and future works of the proposed algorithm.

### 3. Inspiration

#### 3.1. Young's double-slit experiment

Young's Double-Slit Experiment (YDSE) is one of the most well-known classical experiments in the world of physics, demonstrating the wave nature of light [81,82]. In addition, YDSE played an instrumental role in advancing optics and quantum mechanics [82,83]. In earlier times, before the nineteenth century, Isaac Newton was one of the famous scientists who studied light and came up with his theory of light which is known as "corpuscular theory". To explain reflection and refraction, Newton postulated that light is composed of small particles emitted in straight lines until it is refracted from a solid surface [84]. At the same time, when Newton believed in the corpuscular nature of light, his contemporary Christian Huygens proposed that the light proceeds in a wave-like pattern [85]. Fig. 2 depicts Huygens' principle, where the expanding waves may be demonstrated in a ripple tank by sending plane waves towards a barrier with a small opening slit. According to Huygens' principle, every point on the wavefront will act as a secondary source and will be able to emit new secondary waves in all directions [86,87]. These new secondary waves operate effectively in the forward direction with the speed of the wave.

Huygens' theory explained refraction, diffraction, and the interference pattern made as a result. Using an interference pattern, Huygens demonstrated that the shadow edges are not entirely sharp, and consequently, the light is a wave. Both Newton and Huygens performed profound studies and came up with their theories. Despite the

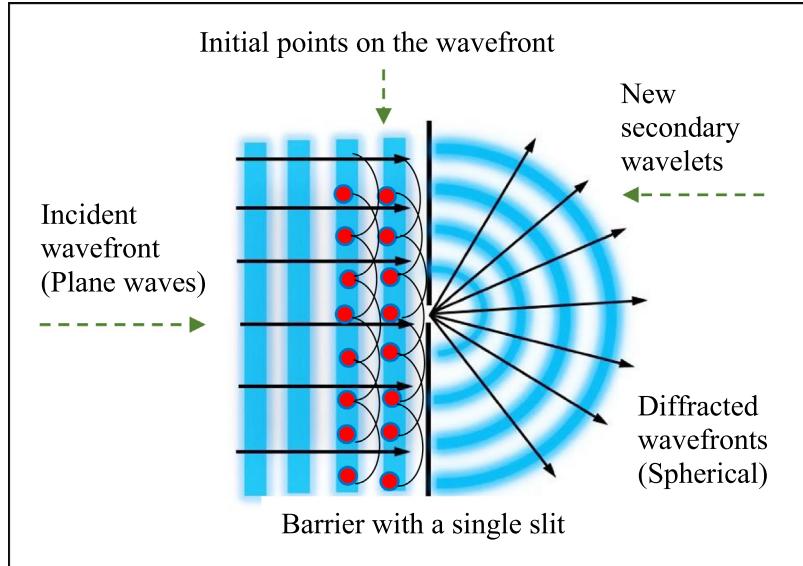


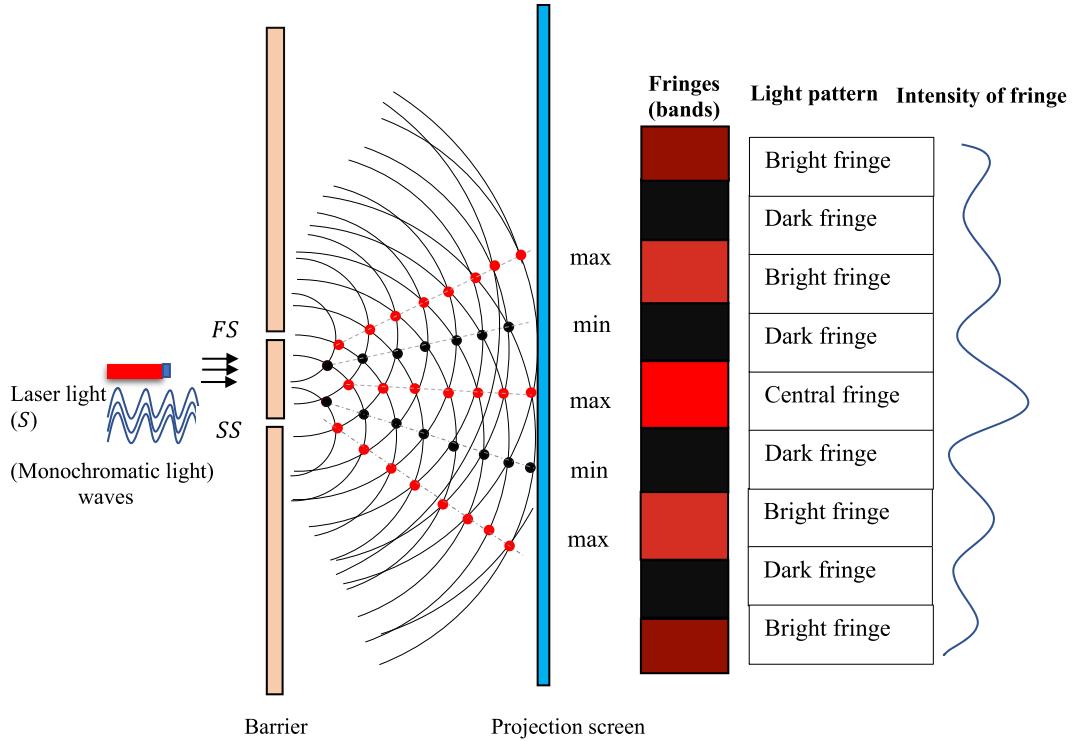
Fig. 2. Huygens' principle.

Monochromatic light	Coherent light
Laser	Single wavelength
	Crest is over crest and trough is over trough
Polychromatic light	Incoherent light
Light bulb	Different wavelengths
	Crests and troughs are not aligned

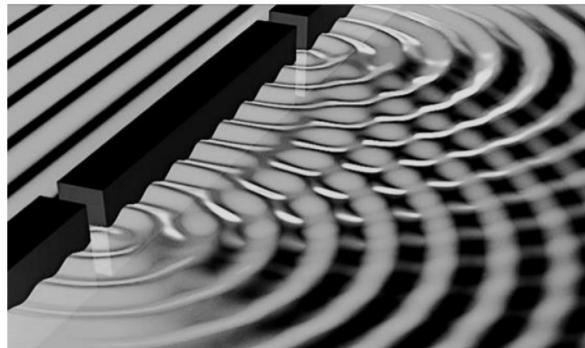
Fig. 3. Illustration of the monochromaticity and coherency concepts.

value of Huygens' theory, it was disregarded by most scientists. Newton's corpuscular theory was widely accepted due to the enormous prestige given to Newton, who could explain the planets' motions through the law of universal gravitation [88]. Later, Thomas Young conducted a double-slit experiment that enormously boosted the wave nature of light. He proved that the interference of light and Huygens' wave theory of light were correct [89]. Subsequently, this experiment was named after him, "Young's double-slit experiment".

Scientists have been puzzled by the phenomenon of light wave interference for many years, but YDSE has been given credit for interpreting this mystery. This experiment used a monochromatic light defined as having a single wavelength. In Fig. 3, we illustrate the concept of monochromaticity and coherency. Most lights like fire and light bulbs are non-monochromatic and incoherent in contrast to lasers that are used as a source of monochromatic and

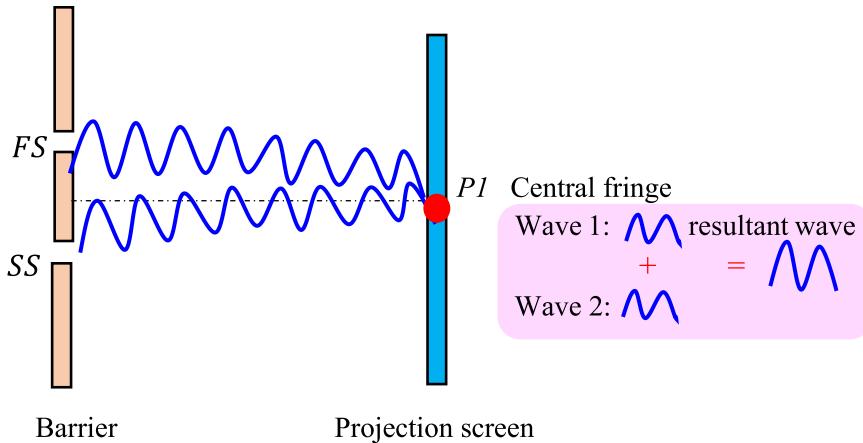
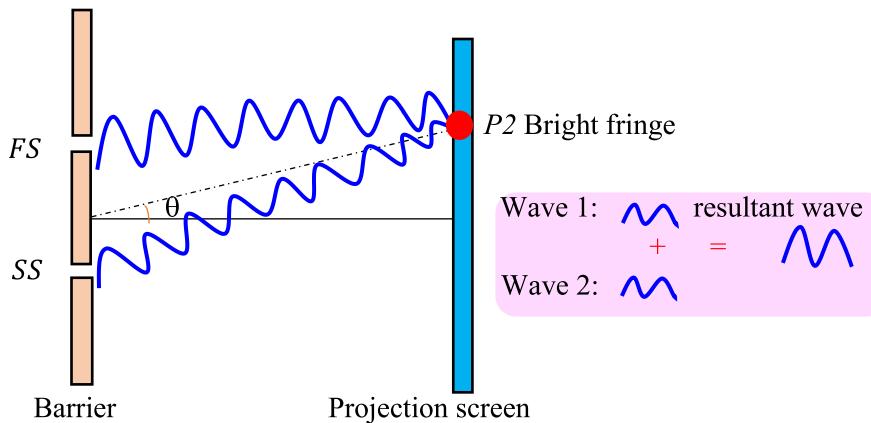


**Fig. 4.** Young's double-slit experiment for the light interference.



**Fig. 5.** Water and light having wave-like pattern.

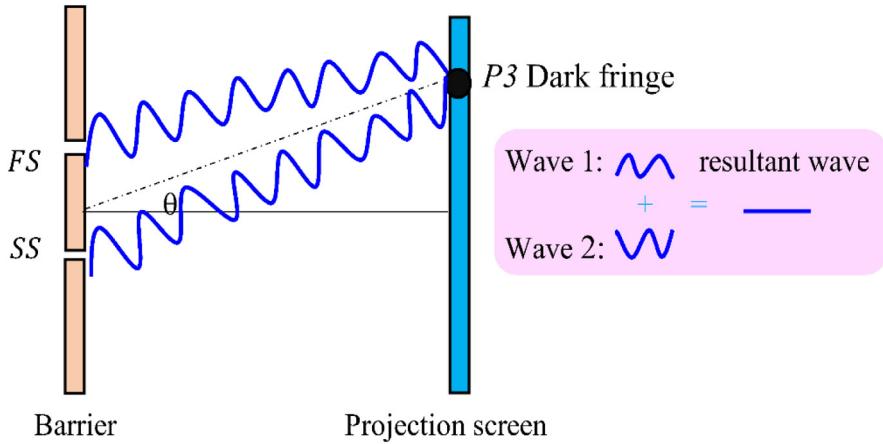
coherent light [90]. Fig. 4 depicts a simple scheme for YDSE. As shown in the figure, YDSE is very simple and is based on the principle of shining light from a monochromatic light source (*S*) on a barrier containing two small slits: the first slit (*FS*) and the second slit (*SS*). Additionally, a projection screen is placed after the barrier showing the end of the light path. Moreover, *FS* and *SS* behave as two coherent sources as they are derived from *S*. The light waves emerging from the two narrow slits interfere and form an interference pattern on the viewing screen. Semicircular waves are formed when light travels through narrow slits. Water waves have a double-slit interference pattern that is remarkably similar to light [91], as in Fig. 5. The bright bands (fringes) correspond to interference maxima, and the dark bands correspond to interference minima. In YDSE, the interference is sustained in which locations for the light intensity of bright and dark fringes stay fixed throughout the screen. Primarily, the two sources must be coherent to obtain a fixed interference pattern.

**Fig. 6.** Constructive interference of central fringe.**Fig. 7.** Constructive interference of bright fringe.

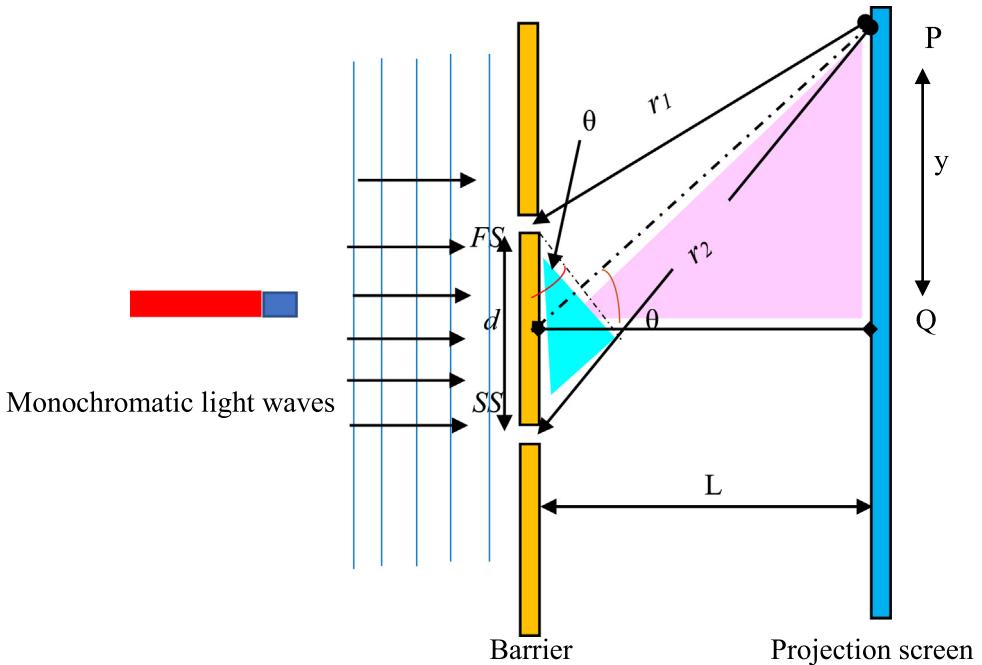
The phenomenon of wave interference happens when two waves moving on the same medium come into contact with one another. There are two sorts of interference: constructive and destructive. Constructive Interference (CI) occurs when the two waves traveling the path from each slit arrive in phase at a point on the screen. Fig. 6 represents a CI where the point  $P_1$  is indicated as a central fringe and the lighting is as strong as possible. Also, Fig. 7 depicts a CI in which the point  $P_2$  is referred to as a bright fringe. Moreover, the lighting is gradually diminishing as this point moves away from the central fringe. In the CI, the resulting wave has a higher amplitude than the interfering waves.

Contrastingly, Destructive Interference (DI) occurs when one wave is longer than the other by half wavelength. Another definition can be when the two waves arrive out of phase at the screen (crest to trough). Fig. 8 represents a DI where the point  $P_3$  is indicated as a dark fringe and no illumination happens in this fringe. From the figure, in the DI, the two interfering waves cancel each other leading to a lower amplitude. Some conditions must be met for the wave interference in YDSE:

- The light source must be monochromatic.
- The waves must be of the same frequency.
- The direction of the waves must be the same.
- The amplitudes of the two waves must be equal.
- The openings of the two slits must be thin.



**Fig. 8.** Destructive interference of dark fringe.



**Fig. 9.** Geometric construction for describing YDSE.

### 3.2. Geometry of YDSE

The geometric structure of YDSE is illustrated in Fig. 9. Assume a monochromatic light source falls on a barrier that contains two slits: FS and SS. A distance  $d$  separates the two slits. The projection screen is placed at a perpendicular distance  $L$  from the barrier. To arrive at any arbitrary point  $P$  in the upper half of the projection screen, the wave coming from the second slit (SS) must move an extra distance greater than the wave coming from the first slit (FS). This extra distance is denoted by  $\Delta L$  and is estimated by the following equation [92–94]:

$$\Delta L = r_2 - r_1 \approx d \sin \theta \quad (1)$$

Pattern of light	Interference	Order number and path difference	Fringe
	Constructive	$m=4, \Delta L=2\lambda$	Bright fringe
	Destructive	$m=3, \Delta L=3/2\lambda$	Dark fringe
	Constructive	$m=2, \Delta L=\lambda$	Bright fringe
	Destructive	$m=1, \Delta L=1/2\lambda$	Dark fringe
	Constructive	$m=0, \Delta L=0$	Central fringe
	Destructive	$m=-1, \Delta L=-1/2\lambda$	Dark fringe
	Constructive	$m=-2, \Delta L=-1\lambda$	Bright fringe
	Destructive	$m=-3, \Delta L=-3/2\lambda$	Dark fringe
	Constructive	$m=-4, \Delta L=-2\lambda$	Bright fringe

Fig. 10. The order number and path difference for the fringes in YDSE.

where  $\Delta L$  determines the path difference and  $\theta$  is the angle shown in Fig. 8.  $r_1$  and  $r_2$  refer to the length of the first wave and the second wave, respectively. Another way to compute  $\Delta L$  [95] depending on the wavelength:

$$\Delta L = \begin{cases} m\lambda, & m = (\pm 0, \pm 2, \pm 4, \dots), \\ (2m+1)\frac{\lambda}{2}, & m = (\pm 1, \pm 3, \pm 5, \dots), \end{cases} \quad \text{if CI} \quad (2)$$

where  $m$  indicates the order number of the interference and  $\lambda$  indicates the wavelength. The CI happens when  $\Delta L$  is an integer multiple of the wavelength ( $\lambda$ ), while the DI happens when the  $\Delta L$  is a half-integral multiple of the wavelength, as illustrated in Fig. 10.

The zeroth order maxima ( $m = 0$ ) correspond to the central bright fringe and the 1st order ( $m = \pm 1$ ) represents the fringes on either side of the central fringe. On the other hand, when  $\Delta L$  is equal to odd integer multiple of  $\lambda/2$ , the waves will be out of phase, resulting in DI with a dark fringe on the screen.

### 3.3. Intensity distribution

The light intensity of the bright fringes differs, where the luminous intensity is strong at the central fringe and gradually decreases as the luminous fringe moves away from the central fringe. In Fig. 4, we can clearly see the gradation of the light intensity of the red color. The light intensity of a fringe can be calculated using [92,96]:

$$Int_P = Int_{\max} \cos^2\left(\frac{\pi d}{\lambda L} y\right) \quad (3)$$

where  $Int_p$  denotes the light intensity of a point  $P$  on the screen, as shown in Fig. 9.  $Int_{\max}$  is the maximum intensity on the screen. The variable  $y$  refers to the distance measured between the central fringe and another fringe at point  $P$ . It is also the position of the bright and the dark fringe which is measured from the central point  $Q$  and is respectively computed as follows [97]:

$$y_{bright} = \frac{\lambda L}{d} m \quad (4)$$

$$y_{dark} = \frac{\lambda L}{d} \left(m + \frac{1}{2}\right) \quad (5)$$

## 4. The proposed algorithm

A turning point in science was reached with Young's double-slit experiment, which proved that light does indeed act like a wave. YDSE optimizer is inspired by the steps of young's double-slit experiment, which demonstrated the wave behavior of light. The light waves displayed interference patterns throughout the experiment. Interference is

the superposition of two or more of coherent oscillating waves in one medium simultaneously on one propagation line. As mentioned in the previous section, there are two types of light interference: CI and DI. CI results in a bright fringe, whereas DI results in a dark fringe at some point on the screen. The following principles derived from this experiment achieve the basic assumptions of our proposed algorithm:

- The source of light must be monochromatic.
- The application of Huygens' principle.
- Interference occurs between two waves only.
- The interference between the two waves is permanent.
- Each fringe resulting from CI or DI represents a possible solution.
- The population size equals the number of fringes (light and dark fringes).
- The fitness of a solution is calculated based on the objective function.

Next, the detailed mathematical formulation of YDSE optimizer will be discussed. Fig. 11 depicts the flowchart of the proposed algorithm.

#### 4.1. Initialization step

According to YDSE, a source of monochromatic light waves ( $S$ ) is first projected into a barrier with two closely spaced slits. Hence, we create an initial monochromatic light source  $S$  consisting of  $NP$  waves as follows:

$$S_{i,j} = Lb_j + \text{rand} \times (Ub_j - Lb_j), \quad (6)$$

$$i = 1, 2, \dots, NP, \quad (7)$$

$$j = 1, 2, \dots, \text{Dim}. \quad (8)$$

$S_{i,j}$  represents the  $j^{th}$  variable of the  $i^{th}$  monochromatic wave.  $Lb_j$  and  $Ub_j$  are respectively the lower and upper bounds of the  $j^{th}$  variable of the problem, and  $\text{rand}$  is a random number in the range of [0,1].

#### 4.2. Huygens' principle

After passing through the two slits, the monochromatic waves spread out in all directions from each slit according to the Huygens' principle. Also, each wavefront point behaves as a source and a center of a new wave. However, for simplicity, we assume that the number of points on the wavefront emerging from the first and second slits is equal to the size of waves  $NP$ , and each point on the wavefront has only one beam. These points can be calculated using the following equations for the wavefront of the two slits:

$$FS_i = S_i + L \times \text{rand1}(-1, 1) \times (S_{mean} - S_i), \quad i = 1, 2, \dots, NP \quad (9)$$

$$SS_i = S_i - L \times \text{rand2}(-1, 1) \times (S_{mean} - S_i), \quad i = 1, 2, \dots, NP \quad (10)$$

where  $FS_i$  is the point  $i$  created on the wavefront outgoing from the first slit  $FS$ , and  $SS_i$  is the point  $i$  created on the wavefront outgoing from the second slit  $SS$ .  $S_{best}$  is the best individual in the population  $S$ .  $S_{mean}$  determines the mean of the current population  $S$  and it is calculated using:

$$S_{mean} = \frac{1}{NP} \sum_{i=1}^{NP} S_i \quad (11)$$

$S_i$  indicates the  $i^{th}$  monochromatic wave that will be emitted through slits. Furthermore,  $\text{rand1}(-1, 1)$  and  $\text{rand2}(-1, 1)$  are uniformly distributed random numbers within the range [-1,1], and  $L$  is the distance between the light source and the barrier.  $\text{rand1}(-1, 1)$  and  $\text{rand2}(-1, 1)$  simulate the scattering of light rays in different directions.

In the YDSE optimizer, the initial state of the creation of the population exactly simulates the initial state to create the first interference of waves in the YDSE. Each bright fringe (CI) and each dark fringe (DI) represents a candidate solution in the search space. In the first generation, the first fringes (bright and dark) are generated using Eqs. (9) and (10). These two equations simulate the first CI (marked with a red circle) and the first DI (marked

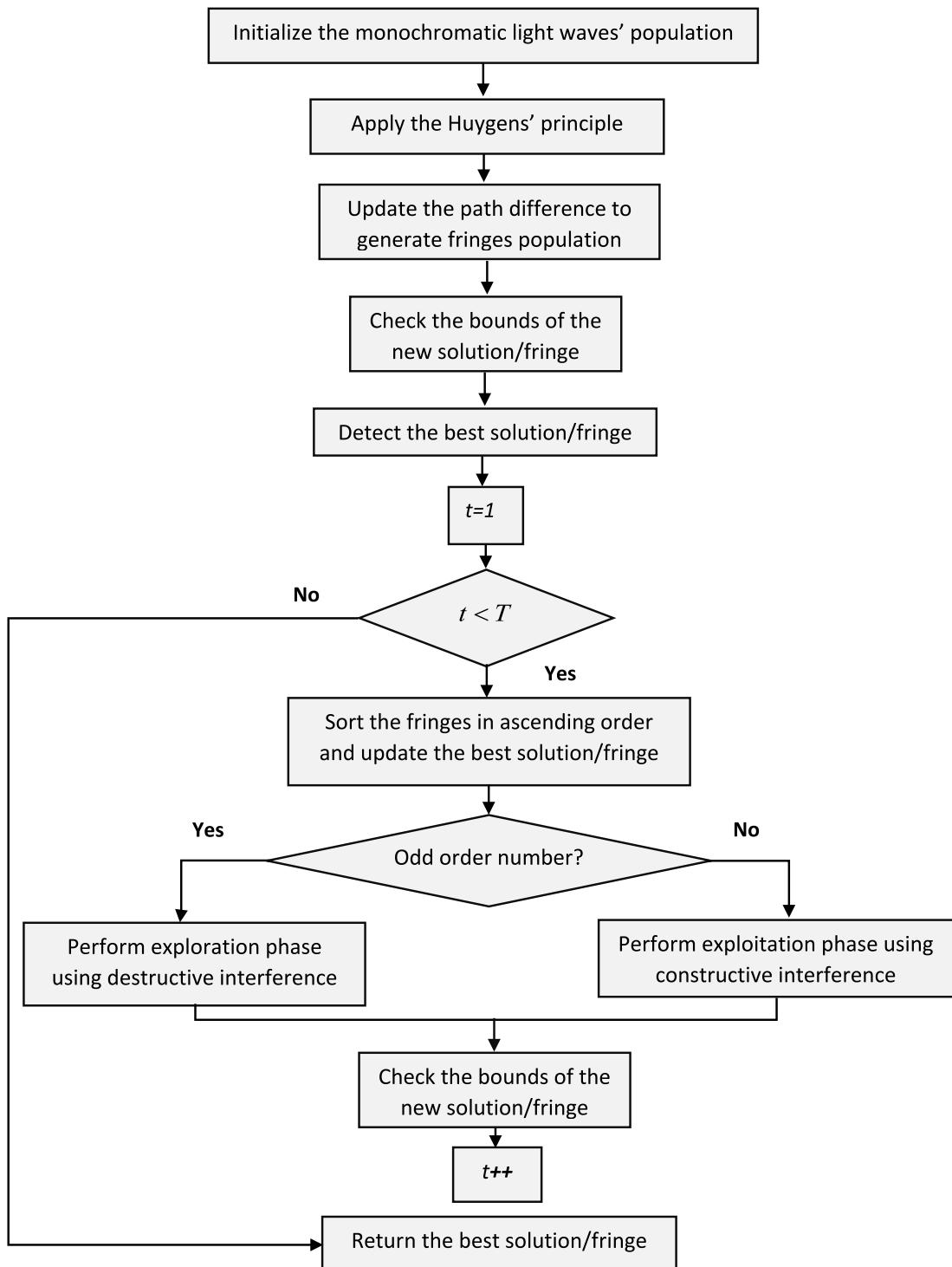


Fig. 11. The flowchart of the proposed YDSE optimizer.

**Table 2**

Index mapping of wavefront points into order numbers.

Index ( $i$ )	Order number ( $m$ )	Type of fringe
1	0	Central
2	1	Dark
3	2	Bright
4	3	Dark
:	:	:
$NP$	$NP - 1$	...

with a black circle) (see Fig. 4). In YDSE, the first interference of the waves occurs after monochromatic light falls on the two-slit barrier, FS and SS. In the YDSE optimizer, Eq. (6) simulates the fall of monochromatic light on a two-slit barrier. For simplicity, each wave (or particle) represents a possible solution, and each fringe resulting from CI or DI represents also a possible solution in the same iteration. In contrast, each solution is evaluated based on the objective function. The convergence behavior of the proposed algorithm and finding a near-optimal solution are described in Section 4.6.

#### 4.3. Traveling waves and path difference update in YDSE optimizer

In this step, the outgoing waves from the two slits did not travel the same distance as one wave from one slit may travel a distance that can be larger than, smaller than or equal to the other wave to reach a point on the screen. The interference patterns are constructed based on points created on the two wavefronts outgoing from the two slits:  $FS$  and  $SS$ . These points are generated using Eqs. (9) and (10). In YDSE, the points of constructive and destructive interference result in bright and dark fringes on the projection screen, respectively. Bright and dark fringes are arranged to start from the central fringe, which takes an order number equal to zero. The bright fringe takes an even number, while the dark fringe takes an odd number. Table 2 presents the index mapping into order number as well as the corresponding fringe type.

To simulate the behavior of the two interfering waves (CI and DI) and the path that each wave follows to reach the screen, we can mathematically model the position updating strategy as proposed below:

$$X_i = \left( \frac{FS_i + SS_i}{2} \right) + \Delta L \quad (12)$$

$\Delta L$  defines the path difference between  $FS_i$  and  $SS_i$ . It is calculated according to the order number of their formed fringe when reaching the screen. For the zeroth and even order numbers, the constructive occurs, whereas for the odd order number, the destructive interference occurs as follows:

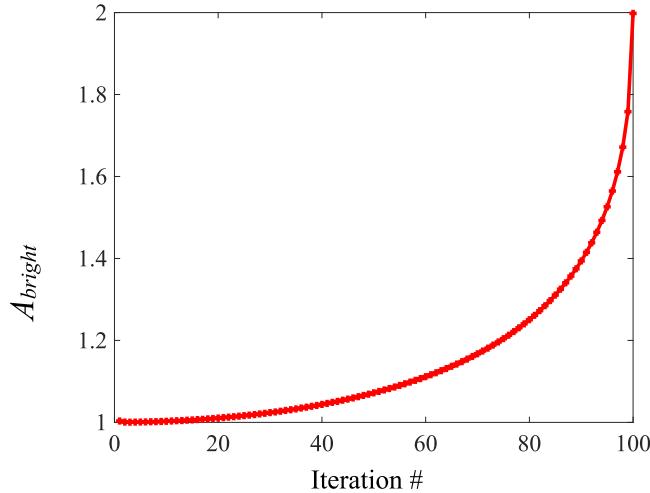
$$\Delta L = \begin{cases} 0 & \text{if } CI \text{ occurs at } m = 0 \\ (2m + 1)\frac{\lambda}{2} & \text{if } DI \text{ occurs at odd } m \\ m\lambda & \text{if } CI \text{ occurs at even } m \end{cases} \quad (13)$$

As can be seen in Table 2, the order number  $m$  is denoted by  $m = i - 1$ . Hence, from Eqs. (12) and (13), the two waves outgoing from two slits have a path difference to reach a point on the screen. When they reach the screen, we start to see some pattern of light called fringes.  $\Delta L$  is zero at the central fringe, which is supposed to be a near-optimal solution in the YDSE algorithm. The farther the fringe is from the central fringe, the large  $\Delta L$  value. In the YDSE optimizer, the further a solution is from the optimal solution, the larger  $\Delta L$  value for that solution.

#### 4.4. Generation of light patterns (fringes)

After the occurrence of the CI and DI, some patterns of light called fringes begin to appear on the projection screen. The population  $X$  generated in Eq. (12) can be viewed as a set of fringes that result from CI and DI (see Fig. 12). The positions of the fringes are fixed as in YDSE, the interference between any two waves is permanent. Therefore, if  $m = 0$ , then  $X_{m=0}$  represents the central fringe. Moreover,  $X_{m_{odd}}$  represents the dark fringe, and  $X_{m_{even}}$

Pattern of light	Order number	Fringe
	$m=0$	Central fringe
	$m=1$	Dark fringe
	$m=2$	Bright fringe
	$m=3$	Dark fringe
...	...	...
	$m$ is odd number	Dark fringe
	$m$ is even number	Bright fringe

**Fig. 12.** Representation for the population of frings.**Fig. 13.** Average of wave amplitude of bright fringe versus iterations.

represents the bright fringe. Each fringe (bright or dark) represents a solution in the search space. The central fringe represents the best solution in the search space. The set of fringes will be optimized through a set of iterations.

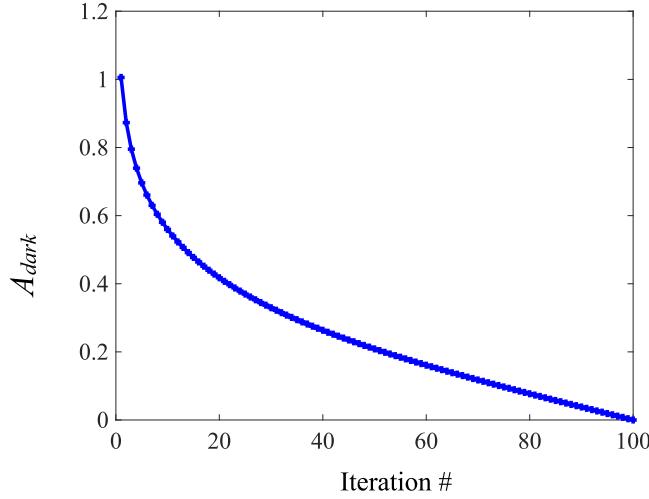
#### 4.5. Wave amplitude update in YDSE optimizer

CI is when two waves superimpose and the resulting wave has a higher amplitude than the previous interfering waves, as illustrated in Figs. 13 and 14. This behavior is mathematically modeled according to this equation:

$$A_{bright}^{t+1} = \frac{2}{1 + \sqrt{|1 - \beta^2|}}, \quad (14)$$

$$\beta = \frac{t}{T} \cosh(\pi/t) \quad (15)$$

$A_{bright}^{t+1}$  indicates the average amplitude of the wave at bright fringe at iteration  $(t + 1)$ .  $t$  refers to the current iteration and  $T$  defines the maximum number of iterations.  $\cosh$  is the hyperbolic function. This equation increases with time and simulates the amplitude of the resultant wave at the bright fringe. In the proposed algorithm, we assume that the average amplitude increases with time and takes the value from 1 at the first iteration to 2 in the last iteration (see Fig. 13). In the other side, DI occurs when two waves superimpose and cancel each other leading



**Fig. 14.** Average of wave amplitude of dark fringe versus iterations.

to a lower amplitude. This behavior is mathematically modeled as follows:

$$A_{dark}^{t+1} = \delta \times \tanh^{-1}\left(-\frac{t}{T} + 1\right) \quad (16)$$

where  $\delta$  is a constant equal to 0.38. This equation simulates the amplitude of the wave at the dark fringe that decreases over iterations. In YDSE optimizer, we assume that the average amplitude decreases with time and takes the value from 1 at the first iteration to 0 in the last iteration (see Fig. 14).

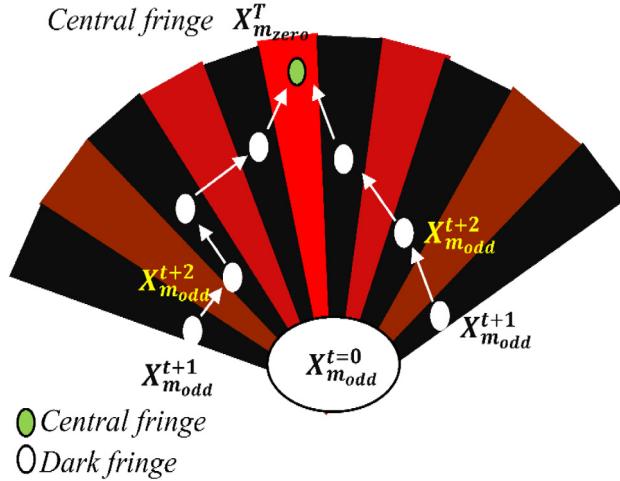
#### 4.6. YDSE optimizer

As shown in Fig. 4, the area of YDSE starts from the light source point to the viewing screen. In addition, the interference starts after the two-slit barrier. This interference is a permanent interference, which means that the intensity positions of bright and dark light remain constant throughout the screen. To simulate this behavior, the search space is visualized in the YDSE optimizer, as shown in Figs. 15–17. These figures show the behavior of solution convergence in the search space. We assume that the search space is divided into three regions: dark regions representing dark fringes; bright regions representing bright fringes; and one central region representing central bright fringes. Figs. 15–17 also depict the central bright region, where a near-optimal solution is expected to exist. In YDSE optimizer, the order number is divided by three particles: even, odd, and 0. If a fringe has an even number, it falls in the light region; if it has an odd number, it falls in the dark region; and if it has zero, it falls in the central bright region. In other words, the order number is related to the position of fringe/particle in search space. This behavior perfectly mimics YDSE to determine locations of fringes. To update the positions in the three regions and to help the agents to move towards the optimal global solution and simultaneously move away from local solutions, the procedures for the exploration and the exploitation stages will be explained in the following subsections.

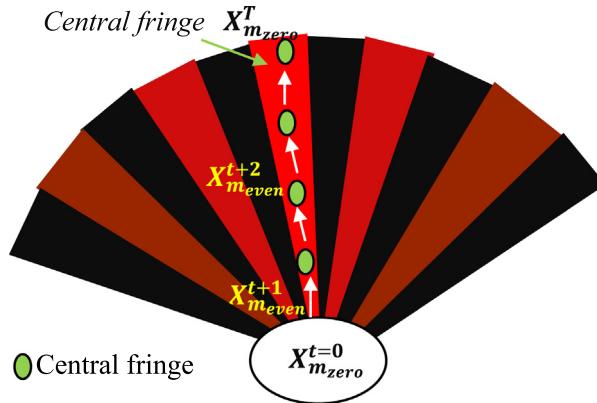
##### 4.6.1. Exploration phase (destructive interference)

During the optimization process, the solution moves in search space based on its order number. If it has an odd number, it moves in the dark regions towards the central bright region that is expected to contain the optimal solution, as illustrated in Fig. 15. At the beginning of the improvement, the solutions in the dark areas are expected to have lower fitness values than those in the light areas. Therefore, to further explore promising positions in dark regions and enhance the YDSE optimizer's ability to avoid falling into the trap of local optima, the following position update strategy is proposed:

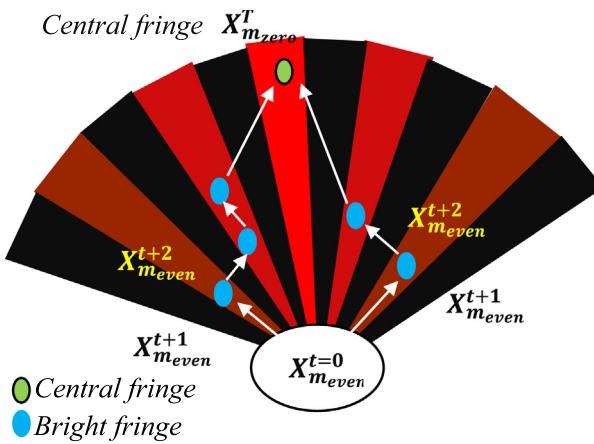
$$X_{m_{odd}}^{t+1} = X_{m_{odd}}^t - (r_1 \times A_{dark}^{t+1} \times Int_{m_{odd}}^{t+1} \times X_{m_{odd}}^t - z \times X_{best}^t) \quad (17)$$



**Fig. 15.** The behavior of solution convergence in the search space for dark fringe.



**Fig. 16.** The behavior of solution convergence in the search space for central fringe.



**Fig. 17.** The behavior of solution convergence in the search space for bright fringe.

where  $X_{m_{odd}}^{t+1}$  is the new  $m_{odd}^{th}$  dark fringe in the iteration  $(t+1)$  and  $X_{m_{odd}}^t$  is the old  $m_{odd}^{th}$  dark fringe in the iteration  $(t)$ .  $r_1$  is a random number in the interval  $[0, 1]$ .  $A_{dark}^{t+1}$  determines the amplitude of the  $m_{odd}^{th}$  dark fringe defined as

in Eq. (16).  $Int_{m_{odd}}^{t+1}$  indicates the light intensity of the  $m_{odd}^{th}$  dark fringe in the iteration ( $t+1$ ) calculated using the next equation.

$$Int_{m_{odd}}^{t+1} = Int_{\max}^{t+1} \times \cos^2\left(\frac{\pi d}{\lambda L} y_{dark}^{t+1}\right) \quad (18)$$

$y_{dark}^{t+1}$  measures the distance between the central fringe and the  $m_{odd}^{th}$  dark fringe which can be computed by Eq. (5).  $Z$  is trial vector of size  $Dim$  defined each iteration as:

$$Z = \frac{a}{H} \quad (19)$$

$$a = t^{2 \times r_2 - 1} \quad (20)$$

where  $a$  takes a random value in  $[T^{-1}, T]$ .  $H$  is a random vector defined in the interval  $[-1, 1]$ , whereas  $r_3$  is a random number in the interval  $[0, 1]$ .  $X_{best}^t$  refers to the best-obtained solution in the iteration ( $t$ ). The vector  $Z$  helps to search around the current best obtained fringe, hoping to find the optimal solution. From Eq. (17), we can see that the new solution moves in the direction of the current dark one, while regarding the difference between both the  $m_{odd}^{th}$  dark fringe and the best solution. At the same time, we follow the dark regions in our search and seek to leave the search around the current best position and explore other different regions of the search space. This strategy helps to get away from the local optima.

#### 4.6.2. Exploitation phase (constructive interference)

In the exploitation phase, the algorithm exploits the promising areas in the bright fringe areas, which are assumed to contain the optimum. YDSE optimizer is working to exploit all the promising areas in bright fringe regions. The update strategy in these regions is formulated as:

$$X_{m_{even}}^{t+1} = X_{m_{even}}^t - ((1 - g) \times A_{bright}^{t+1} \times Int_{m_{even}}^{t+1} \times X_{m_{even}}^t + g \times (Y)) \quad (21)$$

$$Y = X_{m_{rand1}}^t - X_{m_{rand2}}^t \quad (22)$$

where  $X_{m_{even}}^{t+1}$  is the new  $m_{even}^{th}$  bright fringe in the iteration ( $t+1$ ) and  $X_{m_{even}}^t$  is the current  $m_{even}^{th}$  bright fringe in the iteration ( $t$ ).  $g$  is a random number belonging to  $[-1, 1]$ . Additionally,  $A_{bright}^{t+1}$  denotes the resultant wave amplitude at the  $m_{even}^{th}$  bright fringe calculated using Eq. (14).  $Y$  represents the difference between two randomly selected fringes which may be bright, dark or both.  $Int_{m_{even}}^{t+1}$  indicates the intensity of the  $m_{even}^{th}$  bright fringe in iteration  $t+1$  which can be expressed as:

$$Int_{m_{even}}^{t+1} = Int_{\max}^{t+1} \times \cos^2\left(\frac{\pi d}{\lambda L} y_{bright}^{t+1}\right) \quad (23)$$

$y_{bright}^{t+1}$  measures the distance between the central fringe and this  $m_{even}^{th}$  bright fringe.  $Int_{\max}^{t+1}$  represents the maximum intensity detected in the central region in iteration  $t+1$ . Finally, the update strategy in central regions is:

$$X_{m_{zero}}^{t+1} = X_{best}^t + (A_{bright}^{t+1} \times Int_{\max}^{t+1} \times X_{m_{zero}}^t - r_3 \times z \times X_{r_b}^t) \quad (24)$$

$$Int_{\max}^{t+1} = C \times q \quad (25)$$

$$q = \frac{t}{T} \quad (26)$$

where  $X_{m_{zero}}^{t+1}$  determines the new central region which has an order number of zero in the iteration ( $t+1$ ) and  $X_{m_{zero}}^t$  indicates the old position one which has an order number zero in the iteration ( $t$ ). The variable  $r_3$  is a random number  $\in [0, 1]$  raised to the power 5.  $Int_{\max}^{t+1}$  is the maximum intensity detected by the central fringe in the iteration ( $t+1$ ). According to YDSE, the global optimum is a solution that has the maximum intensity value. In the first iterations, the central fringe is far away from the global optimum and hence  $X_{m_{zero}}^{t+1}$  has a lower intensity value. During the iterations, the central fringe gets closer to the global optimum and consequently, the intensity is increased over iterations. Hence, to simulate this behavior,  $q$  is an increasing parameter during iterations from zero to one.  $C$  indicates a constant value equal to  $10^{-20}$  to indicate the maximum intensity value in the last iteration.  $X_{r_b}^t$  represents a bright fringe selected randomly based on  $r_b$  which is an integer number belongs to the even order numbers.

Finally, Eqs. (17), (21) and (24) are combined to be used as follows:

$$X_m^{t+1} = \begin{cases} X_{best}^t + (A_{bright}^{t+1} \times Int_{\max}^{t+1} \times X_m^t - r_3 \times z \times X_{rb}^t), & \text{if } m = 0 \\ X_m^t - ((1 - g) \times A_{bright}^{t+1} \times Int_m^{t+1} \times X_m^t + g \times (Y)), & \text{if } m \text{ is even} \\ X_m^t - (r_1 \times A_{dark}^{t+1} \times Int_m^{t+1} \times X_m^t - Z \times X_{best}^t), & \text{if } m \text{ is odd} \end{cases} \quad (27)$$

Finally, Algorithm 1 presents the pseudocode of the YDSE optimizer.

#### Algorithm 1 YDSE optimizer

**Input:** population size ( $NP$ ), the lower limits of variables ( $Lb$ ), the upper limits of variables ( $Up$ ), size of the problem ( $Dim$ ) the current number of iteration ( $t$ ), the maximum number of iterations ( $T$ ), wavelength ( $\lambda$ ), distance between the light source and the barrier ( $J$ ), distance between the barrier and the screen ( $L$ ) and distance between two slits  $FS$  and  $SS$  ( $d$ ).

**Output:** the best solution  $X_{best}$ .

1. Initialize a monochromatic source of light waves ( $S$ ) using Eq. (6).
2. Apply Huygens' principle to generate the two outgoing wavefronts ( $FS$  and  $SS$ ) from the double-slit using Eq. (9) and Eq. (10).
3. Generate a population  $X$  of  $NP$  light fringes resulting from the outgoing wavefronts and update the path difference using Eq. (12).
4. Check the bounds of the fringes.
5. Evaluate the *fitness* of the solutions/fringes.
6. Detect the best  $X_{best}$  with minimum fitness  $Fitness_{min}$ .
7. Assign an order number for each solution/fringe according to Fig. 12.
8. Define the maximum intensity for the solution /fringe in the central region.
9. Define the current iteration  $t=1$ .
10. **while** ( $t < T$ )
  11. Update  $q$  using Eq. (26)
  12. Sort the fringes from the best to the worst based on their fitness and update  $X_{best}$ .
  13. **for**  $m=0:NP-1$ 
    14. Update  $Z$  using Eq. (19).
    15. **if** ( $m = 0$ ) /\* **Exploitation phase** \*/
      16. Update intensity for the central fringe using Eq. (25).
      17. Update the amplitude for the central fringe using Eq. (14).
      18. Update the central fringe  $X_{m_{zero}}^{t+1}$  using the first state of Eq. (27).
    19. **else if** ( $m = \text{even number}$ )
      20. Update intensity for the bright fringe using Eq. (23).
      21. Update the amplitude for the bright fringe using Eq. (14).
      22. Update the bright fringe  $X_{m_{even}}^{t+1}$  using the second state of Eq. (27).
    23. **else** /\* **Exploration phase** \*/
      24. Update intensity for the dark fringe using Eq. (18).
      25. Update the amplitude for the dark fringe using Eq. (16).
      26. Update the dark fringe  $X_{m_{odd}}^{t+1}$  using the third state of Eq. (27).
    27. **End**
    28. Check the bounds of each fringe
  29. **end for**
  30. Update the current number of iteration  $t$  by  $t=t+1$
  31. **end while**

## 5. The computational complexity of YDSE optimizer

The time complexity of any optimization algorithm depends on its main operations. Hence, YDSE comprises various main operations: initialization, Huygens' principle, path difference update of traveling waves to reach the screen, evaluation process, and solutions updating. The time steps of YDSE can be divided as follows:

- For the *initialization process*, we initialize a set of  $NP$  random solutions each of size  $Dim$ . So, the initialization needs time of  $O(NP \times Dim)$ .

- To apply *the Huygens' principle* and create the outgoing going waves  $FS$  and  $SS$ , it consumes time of  $O(2 \times NP)$ .
- The *path difference update* of the traveling waves to reach a point on the screen takes a time of  $O(NP)$ .
- For the *evaluation process*, since we evaluate  $NP$  solutions based on the cost function ( $C$ ). Thus, the evaluation process after initialization takes  $O(C \times NP)$ . After that, the new solutions are evaluated during a course of times and consumes time of  $O(T \times C \times NP)$ . The overall evaluation process takes  $O(C \times NP) + O(T \times C \times NP)$ .
- The *solutions updating process*, the exploitation, and the exploration phases of YDSE take  $O(\frac{1}{2} \times T \times NP \times Dim) + O(\frac{1}{2} \times T \times NP \times Dim)$ .

The overall time complexity of YDSE is calculated as:

$$\begin{aligned}
Time\ complexity\ (YDSE) &= O(initialization) + O(Huygens'\ principle) + \\
&O(path\ difference\ update) + O(evaluation) + O(solutions\ up\ dating) \\
&= O(NP \times Dim) + O(2 \times NP) + O(NP) + O(C \times NP) + O(T \times C \times NP) \\
&\quad + O(\frac{1}{2} \times T \times NP \times Dim) + O(\frac{1}{2} \times T \times NP \times Dim) \\
&= O(T \times C \times NP) + O(T \times NP \times Dim) \\
&= O(T \times C \times NP + T \times NP \times Dim).
\end{aligned}$$

## 6. The experimental results discussion and analysis

In this section, the results of the proposed algorithm will be shown and analyzed and YDSE will be compared against a wide variety of the existing MH algorithms. In addition, this section has several subsections, which may be summed up as follows:

- Experimental environment setup.
- Characteristics of the test functions.
- Parameter settings of the algorithms.
- Results for CEC 2014 test suite.
- Results for CEC 2017 test suite.
- Results for CEC 2022 test suit.
- Comparison with CEC winners.
- YDSE optimizer for constrained engineering problems.

### 6.1. Experimental environment setup

This subsection will examine the efficacy the YDSE optimizer through various experiments. We run all the experiments on a laptop specified by MacBook Air M1 with chip 8-core CPU and 8-core GPU. The size of the main memory is 8 GB. All the algorithms are implemented under the MATLAB platform.

### 6.2. Characteristics of the test functions

To investigate the efficacy of YDSE, we carried out many experiments. The first experiment is the CEC 2014 [98] that is widely employed for examining the performance of MHs. The second experiment is conducted on the CEC 2017 benchmark [99]. The third experiment mainly includes twelve test functions belonging to the CEC 2022 benchmark [100]. Tables 3–5 list the specifications of the test functions for the CEC 2014, CEC 2017, and CEC 2022, respectively. Each test function is described by its name, dimension ( $Dim$ ), lower and upper bounds [ $lb$ ,  $ub$ ], and the global minimum ( $F_{min}$ ). The CEC 2014 test functions are categorized into four classes: unimodal (F1–F3), multimodal (F4–F16), hybrid (F17–F22), and composition (F23–F30). Moreover, The CEC 2017 test functions are categorized into four classes: unimodal (F1–F3), multimodal (F4–F11), hybrid (F12–F20), and composition (F21–F30). For the CEC 2022 benchmark, F1 belongs to the unimodal test function, F2–F5 belong to the unimodal test functions, F6–F8 belong to the hybrid functions and F9–F12 belong to the composite functions.

**Table 3**

Description of CEC 2014 test suite.

Function type	No.	Name	Dim	[lb, ub]	F <sub>min</sub>
Unimodal function	F1	<i>Rotated high conditioned elliptic function</i>	10	[-100,100]	100
	F2	<i>Rotated Bent cigar function</i>	10	[-100,100]	200
	F3	<i>Rotated discus function</i>	10	[-100,100]	300
Multimodal functions	F4	<i>Shifted and rotated Rosenbrock function</i>	10	[-100,100]	400
	F5	<i>Shifted and rotated Ackley's function</i>	10	[-100,100]	500
	F6	<i>Shifted and rotated Weierstrass function</i>	10	[-100,100]	600
	F7	<i>Shifted and rotated Griewank's function</i>	10	[-100,100]	700
	F8	<i>Shifted Rastrigin function</i>	10	[-100,100]	800
	F9	<i>Shifted and rotated Rastrigin function</i>	10	[-100,100]	900
	F10	<i>Shifted Schwefel function</i>	10	[-100,100]	1000
	F11	<i>Shifted and rotated Schwefel function</i>	10	[-100,100]	1100
	F12	<i>Shifted and rotated Katsuura function</i>	10	[-100,100]	1200
	F13	<i>Shifted and rotated happycat function</i>	10	[-100,100]	1300
	F14	<i>Shifted and rotated hgbat function</i>	10	[-100,100]	1400
	F15	<i>Shifted and rotated expanded Griewank's plus Rosenbrock's function</i>	10	[-100,100]	1500
	F16	<i>Shifted and rotated expanded Scaffer's function</i>	10	[-100,100]	1600
Hybrid functions	F17	<i>Hybrid function 1</i>	10	[-100,100]	1700
	F18	<i>Hybrid function 2</i>	10	[-100,100]	1800
	F19	<i>Hybrid function 3</i>	10	[-100,100]	1900
	F20	<i>Hybrid function 4</i>	10	[-100,100]	2000
	F21	<i>Hybrid function 5</i>	10	[-100,100]	2100
	F22	<i>Hybrid function 6</i>	10	[-100,100]	2200
Composition functions	F23	<i>Composition function 1</i>	10	[-100,100]	2300
	F24	<i>Composition function 2</i>	10	[-100,100]	2400
	F25	<i>Composition function 3</i>	10	[-100,100]	2500
	F26	<i>Composition function 4</i>	10	[-100,100]	2600
	F27	<i>Composition function 5</i>	10	[-100,100]	2700
	F28	<i>Composition function 6</i>	10	[-100,100]	2800
	F29	<i>Composition function 7</i>	10	[-100,100]	2900
	F30	<i>Composition function 8</i>	10	[-100,100]	3000

### 6.3. Parameter settings of the algorithms

The algorithm's performance may be greatly improved by fine-tuning its parameters. As a result, we have tried out various values for all the parameters to see how they affect the performance of the proposed method on different test functions. Our proposed YDSE has four parameters to consider, including Wavelength ( $\lambda$ ), Distance between two slits ( $d$ ), Distance between the barrier and the projection screen ( $L$ ) and Distance between light source and barrier ( $I$ ). As stated by Tipler and Mosca [97], in YDSE, it is assumed that  $L \gg d$  and  $d \gg \lambda$ , such that  $L$  is often the order of 1 m. Consequently,  $L$  is set to 1 m. Also,  $d$  takes a fraction of millimeters ( $\mu\text{m}$ ) and the wavelength takes a fraction of micrometers (mm). We test several values for each parameter to select the best value. For  $\lambda$ , we try several values belong to  $\{1 \times 10^{-6}, 2 \times 10^{-6}, 3 \times 10^{-6}, 4 \times 10^{-6}, 5 \times 10^{-6}, 6 \times 10^{-6}, 7 \times 10^{-6}, 8 \times 10^{-6}, 9 \times 10^{-6}\}$ , whereas  $d$  is tested with values belonging to  $\{1 \times 10^{-3}, 2 \times 10^{-3}, 3 \times 10^{-3}, 4 \times 10^{-3}, 5 \times 10^{-3}, 6 \times 10^{-3}, 7 \times 10^{-3}, 8 \times 10^{-3}, 9 \times 10^{-3}\}$ . Furthermore,  $I$  is examined by a value from several values, including  $\{0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09\}$ . Several experiments were conducted to choose the optimal values of three parameters by trial and error on different types of the test functions. From the experiments, we conclude that the optimal values for YDSE parameters that obtain the best results as follows:  $\lambda = 5 \times 10^{-6}$ ,  $d = 5 \times 10^{-3}$  and  $I = 0.01$ . Additionally, the population size is limited to 30.

In order to have a better idea of how successful YDSE is as a whole, we compare YDSE with other eleven well-known MH algorithms that fall into one of the five categories of MHs that are outlined as follows:

1. Differential Evolution (DE) [22].
2. Multi-Trial vector-based Differential Evolution (MTDE) [23].

**Table 4**

Description of CEC 2017 test suite.

Function type	No.	Name	Dim	[lb, ub]	F <sub>min</sub>
Unimodal function	F1	<i>Shifted and rotated Bent cigar function</i>	10	[-100,100]	100
Multimodal functions	F3	<i>Shifted and rotated Rosenbrock's function</i>	10	[-100,100]	300
	F4	<i>Shifted and rotated Rastrigin's function</i>	10	[-100,100]	400
	F5	<i>Shifted and rotated Expanded Scaffer's F6 function</i>	10	[-100,100]	500
	F6	<i>Shifted and rotated lunacek Birastrigin function</i>	10	[-100,100]	600
	F7	<i>Shifted and rotated non-continuous Rastrigin's function</i>	10	[-100,100]	700
	F8	<i>Shifted and rotated Levy function</i>	10	[-100,100]	800
Hybrid functions	F9	<i>Shifted and rotated Schwefel's function</i>	10	[-100,100]	900
	F10	<i>Hybrid function 1 (N = 3)</i>	10	[-100,100]	1000
	F11	<i>Hybrid function 2 (N = 3)</i>	10	[-100,100]	1100
	F12	<i>Hybrid function 3 (N = 3)</i>	10	[-100,100]	1200
	F13	<i>Hybrid function 4 (N = 4)</i>	10	[-100,100]	1300
	F14	<i>Hybrid function 5 (N = 4)</i>	10	[-100,100]	1400
	F15	<i>Hybrid function 6 (N = 4)</i>	10	[-100,100]	1500
	F16	<i>Hybrid function 7 (N = 5)</i>	10	[-100,100]	1600
	F17	<i>Hybrid function 8 (N = 5)</i>	10	[-100,100]	1700
	F18	<i>Hybrid function 9 (N = 5)</i>	10	[-100,100]	1800
Composite functions	F19	<i>Hybrid function 10 (N = 6)</i>	10	[-100,100]	1900
	F20	<i>Composite function 1 (N = 3)</i>	10	[-100,100]	2000
	F21	<i>Composite function 2 (N = 3)</i>	10	[-100,100]	2100
	F22	<i>Composite function 3 (N = 4)</i>	10	[-100,100]	2200
	F23	<i>Composite function 4 (N = 4)</i>	10	[-100,100]	2300
	F24	<i>Composite function 5 (N = 5)</i>	10	[-100,100]	2400
	F25	<i>Composite function 6 (N = 5)</i>	10	[-100,100]	2500
	F26	<i>Composite function 7 (N = 6)</i>	10	[-100,100]	2600
	F27	<i>Composite function 8 (N = 6)</i>	10	[-100,100]	2700
	F28	<i>Composite function 9 (N = 6)</i>	10	[-100,100]	2800
	F29	<i>Composite function 10 (N = 3)</i>	10	[-100,100]	2900
	F30	<i>Composite function 11 (N = 3)</i>	10	[-100,100]	3000

**Table 5**

Description of CEC 2022 test suite.

Function type	No.	Name	Dim	[lb, ub]	F <sub>min</sub>
Unimodal function	F1	<i>Shifted and full rotated Zakharov function</i>	10, 20	[-100,100]	300
Basic functions	F2	<i>Shifted and full rotated Rosenbrock's function</i>	10, 20	[-100,100]	400
	F3	<i>Shifted and full rotated expanded Schaffer's F6 function</i>	10, 20	[-100,100]	600
	F4	<i>Shifted and full rotated non-continuous Rastrigin's function</i>	10, 20	[-100,100]	800
	F5	<i>Shifted and rotated Levy function</i>	10, 20	[-100,100]	900
Hybrid functions	F6	<i>Hybrid function 1 (N = 3)</i>	10, 20	[-100,100]	1800
	F7	<i>Hybrid function 2 (N = 6)</i>	10, 20	[-100,100]	2000
	F8	<i>Hybrid function 3 (N = 5)</i>	10, 20	[-100,100]	2200
Composite functions	F9	<i>Composite function 1 (N = 5)</i>	10, 20	[-100,100]	2300
	F10	<i>Composite function 2 (N = 4)</i>	10, 20	[-100,100]	2400
	F11	<i>Composite function 3 (N = 5)</i>	10, 20	[-100,100]	2600
	F12	<i>Composite function 4 (N = 6)</i>	10, 20	[-100,100]	2700

3. Particle Swarm Optimization (PSO) [30].
4. African Vulture Optimization Algorithm (AVOA) [41].
5. Whale Optimization Algorithm (WOA) [42].
6. Slime Mould Algorithm (SMA) [49].
7. Sine Cosine Algorithm (SCA) [57].
8. Coronavirus Herd Immunity Optimizer (CHIO) [79].

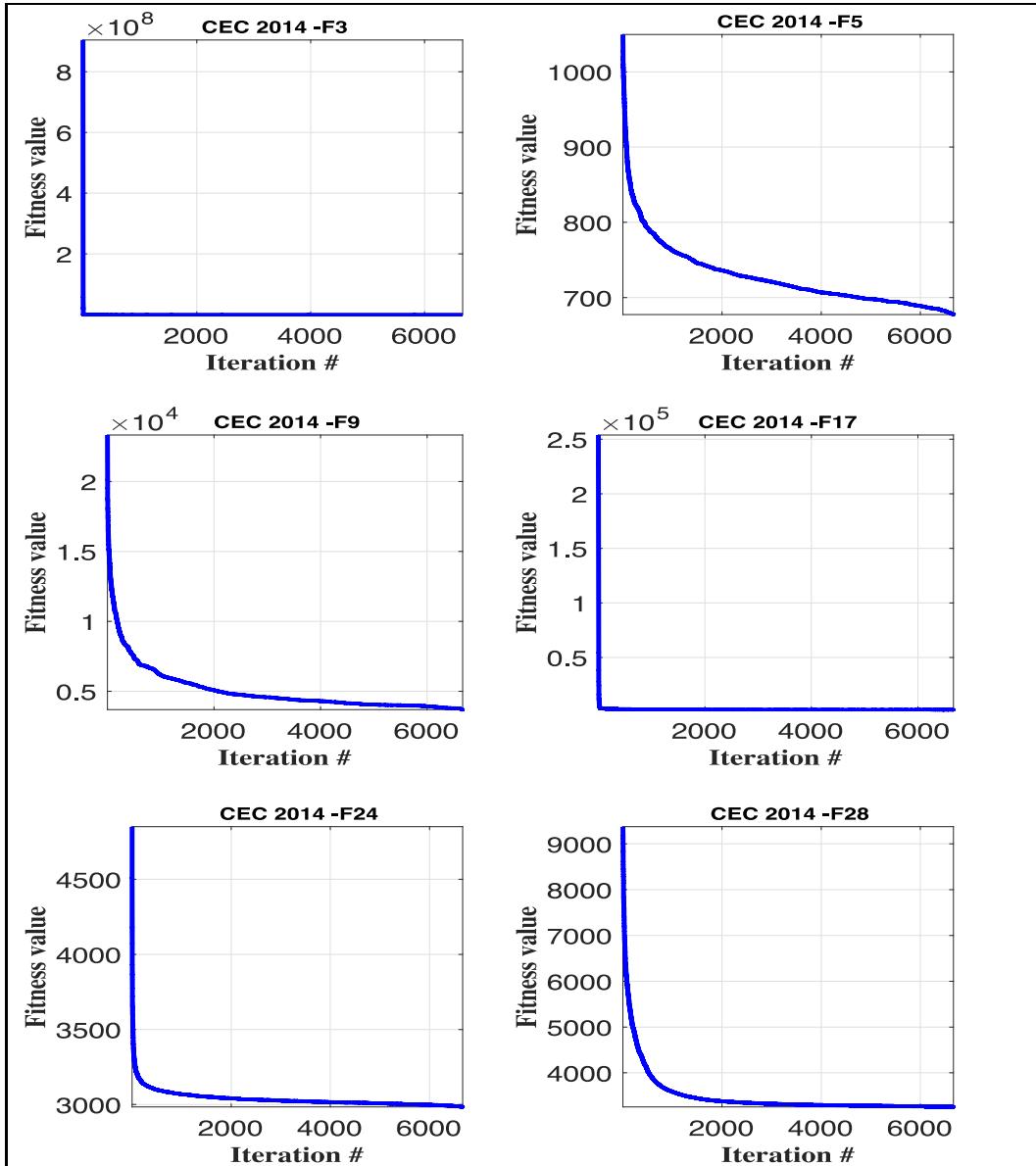
**Table 6**

The parameter settings of the optimization algorithms.

Algorithm	Parameter	Value
YDSE	Wavelength ( $\lambda$ )	$5 \times 10^{-6}$ m
	Distance between two slits ( $d$ )	$5 \times 10^{-3}$ m
	Distance between the barrier and the projection screen ( $L$ )	1 m
	Distance between light source and barrier ( $I$ )	0.01 m
	Constant value ( $\delta$ )	0.38
	Population size ( $NP$ )	30
DE [22]	Mutation rate	0.5
	Crossover rate	0.5
	Population size ( $NP$ )	50
MTDE [23]	Mutation factor	0.5
	Population size ( $NP$ )	20
	WinIter	20
	H	5
	Initial	0.001
	Final	2
PSO [30]	$\sigma$	0.2
	Mu	Log ( $Dim$ )
	Cognitive coefficient	2
	Social coefficient	2
AVOA [41]	Inertia constant	Decreased in [0.9–0]
	Population size ( $NP$ )	50
	Population size ( $NP$ )	30
WOA [42]	$L_1$	Low = 0.9, medium = 0.8, high = 0.7
	$L_2$	Low = 0.1, medium = 0.2, high = 0.3
	w	Low = 2, medium = 2.5, high = 3
	$P_1$	Low = 0.4 medium = 0.5 high = 0.6
	$P_2$	Low = 0.4, medium = 0.5, high = 0.6
	$P_3$	Low = 0.4, medium = 0.5, high = 0.6
SMA [49]	Convergence parameter (a)	Reduced in [2, 0]
	r	[−1, 1]
SCA [57]	Population size ( $NP$ )	30
	Population size ( $NP$ )	30
	z	0.03
CHIO [79]	Population size ( $NP$ )	30
	Constant value (a)	2
RSO [101]	Population size ( $NP$ )	30
	BR <sub>r</sub>	0.01
HGS [102]	Population size ( $NP$ )	30
	Control parameter (R)	[1, 5]
	Constant parameter (C)	[0, 2]
WSO [103]	Population size ( $NP$ )	30
	l	0.08
WSO [103]	Hanger threshold (LH)	10 000
	Population size ( $NP$ )	30

9. Rat Swarm Optimizer (RSO) [101].
10. Hunger Games Search (HGS) algorithm [102].
11. War Strategy Optimization (WSO) algorithm [103].

Both DE and MTDE are members of the family of EAs. On the other hand, PSO, WOA, RSO, and AVOA are all considered to be SI algorithms. HGS, WSO, and CHIO are derived from HAs, but SCA and SMA have a physical basis for their existence. In addition, the parameters of these MHs are figured out in line with the suggestions stated by the authors of their initial publications. The parameter settings for each MH are listed in [Table 6](#). Each algorithm is evaluated by thirty times.



**Fig. 18.** The convergence curve of YDSE over some CEC 2014 test functions.

#### 6.4. Results of CEC 2014 test suite

Here, we provide the analysis of the YDSE optimizer performed on the CEC 2014 benchmark. This benchmark consists of 30 test functions and their specifications are recorded in Table 3. Fig. 18 shows the convergence curve of the proposed algorithm for some test functions of the CEC 2014 benchmark (F3, F5, F9, F17, F24, F28). The figure depicts the relation between the mean fitness values through thirty independent runs and the number of iterations. As can be seen, we find that YDSE converges quickly to the optimal solution. Moreover, we compare YDSE with other existing MH algorithms extracted from the literature, such as DE, MTDE, PSO, AVOA, CHIO, HGS, RSO, SCA, SMA, WOA, and WSO. Thirty independent runs assess each algorithm. Also, the number of function evaluations is limited to 200,000 for each algorithm. The other settings of the algorithms are provided in Table 6. This experiment employs two main performance measures: average (Avg) and Standard Deviation (SD). The average is computed

by the summation of all the fitness values obtained by each algorithm during thirty independent runs and divided by 30. Furthermore, the SD is a statistic metric that gives insight into the dispersion of those thirty fitness values of each algorithm relative to its average.

The results of the twelve algorithms are listed in [Tables 7](#) and [8](#). The unimodal test functions are commonly used to evaluate the exploitation capability of an optimization algorithm. As a result, we assess the exploitation behavior of YDSE using these unimodal functions. In the CEC 2014 benchmark, the YDSE optimizer can reach the global optimum on the three unimodal functions from F1 to F3. Also, the multimodal functions are commonly used to evaluate the exploration capability of an optimization algorithm. Hence, the multimodal test functions from F4 to F16 are employed to test the exploration behavior of YDSE. The hybrid and composite functions (F17–F30) are more challenging test functions than the unimodal and multimodal [\[104\]](#). By examining the results of the algorithms, we see that YDSE outperforms its competitors on 20 out of 30 test functions, except for F5, F8, F10, F11, F14, F16, F19, F25, F28, and F29. Meanwhile, our proposed YDSE obtains the minimum SD values in most of the test functions, which means that the fitness values obtained by YDSE are close to the average value. Accordingly, YDSE is in the first position compared to its eleven competitors. Moreover, CHIO obtains the best average values on three test functions (F10, F16, F19, F25, F28), whereas MTDE obtains the best average values on F14 and F29.

For more statistical analyses and significance test for the results of different algorithms, the Wilcoxon signed rank-sum test is employed. The non-parametric Wilcoxon signed rank-sum test is used to indicate if the results of the proposed YDSE algorithm differ statistically from those of existing metaheuristics. In [Table 9](#), a Wilcoxon signed rank-sum test between the YDSE optimizer, and each algorithm is performed with a significance level of  $\alpha = 0.05$  for all the thirty test functions of the CEC 2014 benchmark. Moreover, the  $p$ -values express how the results are significantly different between any two algorithms. The lower  $p$ -value ( $<0.05$ ), the higher proof of a significant difference and the excellence of the proposed algorithm. The bold values indicate the outperformance of the proposed YDSE as  $p$ -values are less than  $\alpha$ . As can be concluded from [Tables 7–9](#), YDSE beats all other algorithms on the vast majority of functions tested with a  $p$ -value  $<0.05$ .

[Figs. 19–22](#) depict the boxplots of the overall fitness values obtained by the algorithms over thirty runs for the CEC 2014 benchmark for the unimodal, multimodal, hybrid and composite functions. The purpose is to investigate the impact of the different function types on the comparing algorithms. From the boxplots, we can deduce five performance metrics: the minimum fitness, the maximum fitness, the first quartile, the third quartile and the median of each method. In most of the test functions, the height of the boxplot is lower than the boxplots of the other algorithms. The median value is determined by the red line inside the box. Based on the figures, YDSE has a lower median and a smaller interquartile range in most of the functions.

[Fig. 23](#) introduces a comparison among the twelve algorithms in terms of the average CPU time on thirty test functions of the CEC 2014 benchmark. As can be seen from the figure, WOA consumes the minimum time value of 0.77 s in average to solve a single CEC 2014 function, whereas PSO consumes the maximum CPU time of 5.99 s. Additionally, YDSE comes in the third rank of CPU time of 0.86 s after WAOA and WSO. The YDSE average CPU time is reasonable when compared with other algorithms like DE, MTDE, PSO, AVOA, CHIO, HGS, RSO, SCA and SMA that take more CPU time.

## 6.5. Results of CEC 2017 test suite

In this experiment, we verify the performance of the suggested algorithm in comparison with its rivals on the widely used CEC 2017 test suite. The CEC 2017 benchmark is regarded as one of the most challenging benchmarks that have been employed for assessing the exploration and exploitation capabilities of the suggested algorithm. We perform a comparison between YDSE and the other algorithms, including DE, MTDE, PSO, AVOA, CHIO, HGS, RSO, SCA, SMA, WOA, and WSO. The CEC 2017 competition specifies a maximum number of runs to be 30 separate runs. Additionally, the number of evaluations for each function is limited to 200,000 for each algorithm. The values of the parameters of each algorithm are set as suggested by their authors, as provided in [Table 6](#). We employ two performance metrics in our comparison: Avg and SD. [Fig. 24](#) depicts the proposed algorithm's convergence curve for certain CEC 2017 benchmark test functions (F1, F4, F10, F16, F21, F29). The graph shows the relationship between the mean fitness values obtained from thirty separate runs and the number of iterations. Looking at functions F4, F16, and F29, we see that YDSE comes close to the optimal solution during the early iterations.

**Table 7**

The results of the different algorithms for CEC 2014 test functions (F1–F15).

Fn.	DE	MTDE	PSO	AVOA	CHIO	HGS	RSO	SCA	SMA	WOA	WSO	YDSE	
F1	Avg	2.0592E+05	1.0000E+02	6.5631E+05	2.5655E+05	3.8314E+05	7.1688E+04	1.0317E+07	2.7748E+06	1.5812E+05	4.7463E+06	1.6370E+04	<b>1.0000E+02</b>
	SD	1.4596E+05	1.1196E−14	1.8993E+06	1.9883E+05	1.9406E+05	7.8556E+04	5.1978E+06	1.1994E+06	1.6072E+05	4.5370E+06	3.8786E+04	<b>0.0000E+00</b>
F2	Avg	9.5522E+03	2.0000E+02	1.1816E+08	9.4062E+03	1.1644E+03	1.2214E+04	2.5567E+09	2.4501E+08	1.0074E+04	2.0120E+04	2.0000E+02	<b>2.0000E+02</b>
	SD	8.4304E+03	1.0556E−14	2.7084E+08	9.2781E+03	1.3161E+03	1.3022E+04	1.8300E+09	1.0242E+08	1.0649E+04	2.2788E+04	1.3589E−02	<b>0.0000E+00</b>
F3	Avg	1.4838E+03	3.0000E+02	9.6001E+02	1.4331E+03	2.3739E+03	1.6888E+03	1.2035E+04	4.0652E+03	3.0227E+02	6.9892E+03	3.5662E+02	<b>3.0000E+02</b>
	SD	9.8138E+02	2.3603E−14	2.5178E+03	1.1071E+03	1.4502E+03	1.0931E+03	4.5355E+03	1.5763E+03	4.6824E+00	3.7042E+03	1.0401E+02	<b>0.0000E+00</b>
F4	Avg	4.0332E+02	4.0000E+02	4.1792E+02	4.0310E+02	4.0114E+02	4.0312E+02	6.3525E+02	4.2713E+02	4.0311E+02	4.1223E+02	4.0049E+02	<b>4.0000E+02</b>
	SD	1.2027E+01	3.9495E−14	3.8868E+01	1.2443E+01	1.3151E+00	1.3168E+01	1.1300E+02	1.0639E+01	8.4526E−01	2.0422E+01	1.1308E+00	<b>0.0000E+00</b>
F5	Avg	5.2002E+02	5.2007E+02	5.2021E+02	5.2003E+02	5.2003E+02	5.2001E+02	5.2028E+02	5.2031E+02	5.2003E+02	5.2005E+02	<b>5.1907E+02</b>	5.2001E+02
	SD	4.0295E−02	2.2178E−02	8.7048E−02	6.2212E−02	9.1338E−03	1.5945E−02	9.1621E−02	6.7892E−02	2.7610E−02	8.1702E−02	4.8776E+00	9.5922E−03
F6	Avg	6.0519E+02	6.0454E+02	6.0116E+02	6.0506E+02	6.0182E+02	6.0383E+02	6.0905E+02	6.0657E+02	6.0371E+02	6.0632E+02	6.0268E+02	<b>6.0082E+02</b>
	SD	1.4921E+00	1.0484E+00	1.4089E+00	1.8780E+00	7.2537E−01	1.5699E+00	1.3249E+00	9.7642E−01	1.2285E+00	1.4378E+00	1.4894E+00	<b>8.4153E−01</b>
F7	Avg	7.0037E+02	7.0005E+02	7.0012E+02	7.0036E+02	7.0003E+02	7.0015E+02	7.3988E+02	7.0434E+02	7.0021E+02	7.0064E+02	7.0068E+02	<b>7.0002E+02</b>
	SD	2.3676E−01	2.9788E−02	4.9943E−02	2.7427E−01	1.7770E−02	7.0938E−02	1.2714E+01	1.1140E+00	7.3467E−02	3.8753E−01	4.0979E−01	<b>1.2229E−02</b>
F8	Avg	8.0464E+02	8.0003E+02	8.0216E+02	8.0547E+02	8.0000E+02	<b>8.0000E+02</b>	8.4233E+02	8.2852E+02	8.0000E+02	8.3015E+02	8.0451E+02	8.0183E+02
	SD	2.7693E+00	1.8165E−01	2.0067E+00	2.8593E+00	5.9838E−04	<b>2.1111E−14</b>	7.7628E+00	4.6253E+00	4.2803E−05	9.9553E+00	3.2186E+00	7.7171E−01
F9	Avg	9.2454E+02	9.0769E+02	9.0660E+02	9.2345E+02	9.1062E+02	9.1459E+02	9.4537E+02	9.3031E+02	9.1141E+02	9.3592E+02	9.0769E+02	<b>9.0517E+02</b>
	SD	1.0005E+01	3.4756E+00	2.8079E+00	8.9477E+00	3.0340E+00	8.2864E+00	6.5196E+00	6.5302E+00	4.8373E+00	1.3164E+01	3.2207E+00	<b>1.9271E+00</b>
F10	Avg	1.1943E+03	1.0086E+03	1.2792E+03	1.2636E+03	<b>1.0002E+03</b>	1.0200E+03	2.1546E+03	1.8009E+03	1.1550E+03	1.6781E+03	1.2180E+03	1.0446E+03
	SD	1.6165E+02	6.9012E+00	1.6732E+02	2.1064E+02	<b>1.1421E−01</b>	4.0236E+01	2.3689E+02	1.9727E+02	8.2788E+01	3.1665E+02	1.1566E+02	3.8090E+01
F11	Avg	1.9002E+03	1.3967E+03	1.4945E+03	1.9485E+03	1.3965E+03	1.5959E+03	2.3261E+03	2.3690E+03	1.7786E+03	2.1320E+03	<b>1.3654E+03</b>	1.5072E+03
	SD	2.8887E+02	1.6678E+02	1.8261E+02	2.5173E+02	1.0071E+02	2.1374E+02	3.0673E+02	1.5701E+02	2.0108E+02	2.9229E+02	1.6011E+02	1.6281E+02
F12	Avg	1.2004E+03	1.2003E+03	1.2001E+03	1.2004E+03	1.2002E+03	1.2001E+03	1.2008E+03	1.2009E+03	1.2001E+03	1.2007E+03	1.2011E+03	<b>1.2001E+03</b>
	SD	2.0265E−01	1.6376E−01	1.1196E−01	2.2649E−01	5.0721E−02	1.0308E−01	2.8286E−01	1.7872E−01	9.1476E−02	2.7540E−01	1.9363E−01	<b>5.8684E−02</b>
F13	Avg	1.3004E+03	1.3001E+03	1.3001E+03	1.3004E+03	1.3002E+03	1.3005E+03	1.3009E+03	1.3005E+03	1.3002E+03	1.3005E+03	1.3002E+03	<b>1.3001E+03</b>
	SD	1.5919E−01	5.1939E−02	6.6267E−02	1.4664E−01	5.7365E−02	1.6019E−01	3.0300E−01	1.0354E−01	6.5310E−02	1.6780E−01	1.3725E−01	<b>2.3859E−02</b>
F14	Avg	1.4003E+03	<b>1.4001E+03</b>	1.4005E+03	1.4003E+03	1.4002E+03	1.4004E+03	1.4123E+03	1.4008E+03	1.4001E+03	1.4003E+03	1.4002E+03	1.4001E+03
	SD	1.3421E−01	<b>1.9608E−02</b>	3.1919E−01	1.0999E−01	4.4007E−02	2.2297E−01	7.4590E+00	2.9286E−01	4.5876E−02	2.3698E−01	1.8195E−01	3.1146E−02
F15	Avg	1.5032E+03	1.5006E+03	1.5026E+03	1.5033E+03	1.5013E+03	1.5015E+03	1.0831E+04	1.5089E+03	1.5009E+03	1.5107E+03	1.5016E+03	<b>1.5006E+03</b>
	SD	2.1031E+00	3.0699E−01	7.1988E+00	1.7582E+00	4.2058E−01	9.8591E−01	3.9392E+04	1.5653E+00	3.2243E−01	6.7900E+00	1.0614E+00	<b>1.6916E−01</b>

The bold font indicates the best results.

**Table 8**

The results of the different algorithms for CEC 2014 test functions (F16–F30).

Fn.	DE	MTDE	PSO	AVOA	CHIO	HGS	RSO	SCA	SMA	WOA	WSO	YDSE
F16	Avg	$1.60E+03$	$1.60E+03$	$1.60E+03$	$1.60E+03$	<b><math>1.60E+03</math></b>	$1.60E+03$	$1.60E+03$	$1.60E+03$	$1.60E+03$	$1.60E+03$	$1.60E+03$
	SD	$3.36E-01$	$6.63E-01$	$7.75E-01$	$3.42E-01$	<b><math>2.36E-01</math></b>	$4.71E-01$	$2.38E-01$	$1.38E-01$	$4.55E-01$	$2.52E-01$	$4.97E-01$
F17	Avg	$2.05E+03$	$1.70E+03$	$4.50E+04$	$2.01E+03$	$8.97E+03$	$1.78E+03$	$7.62E+04$	$7.27E+03$	$1.78E+03$	$1.07E+05$	$1.77E+03$
	SD	$2.19E+02$	$4.00E+00$	$2.14E+05$	$2.00E+02$	$8.05E+03$	$6.39E+01$	$3.89E+04$	$1.89E+03$	$8.80E+01$	$2.35E+05$	$7.82E+01$
F18	Avg	$7.68E+03$	$1.80E+03$	$6.71E+03$	$6.90E+03$	$3.44E+03$	$1.01E+04$	$8.30E+03$	$5.62E+04$	$1.34E+04$	$6.41E+03$	$1.83E+03$
	SD	$6.54E+03$	$8.81E-01$	$7.39E+03$	$6.04E+03$	$1.36E+03$	$7.47E+03$	$9.75E+03$	$5.05E+04$	$7.77E+03$	$5.20E+03$	$3.41E+01$
F19	Avg	$1.90E+03$	$1.90E+03$	$1.90E+03$	$1.90E+03$	<b><math>1.90E+03</math></b>	$1.90E+03$	$1.92E+03$	$1.90E+03$	$1.90E+03$	$1.91E+03$	$1.90E+03$
	SD	$2.15E+00$	$9.59E-02$	$1.28E+00$	$2.30E+00$	$3.54E-01$	$1.76E+00$	$2.17E+01$	$7.13E-01$	$1.29E+00$	$2.74E+00$	$7.02E-01$
F20	Avg	$2.04E+03$	$2.00E+03$	$2.55E+03$	$2.04E+03$	$2.16E+03$	$2.02E+03$	$1.59E+04$	$2.34E+03$	$2.01E+03$	$3.25E+03$	$2.01E+03$
	SD	$2.46E+01$	$4.00E-01$	$2.89E+03$	$2.17E+01$	$2.00E+02$	$1.43E+01$	$3.17E+04$	$2.50E+02$	$7.07E+00$	$1.27E+03$	$7.20E+00$
F21	Avg	$6.76E+03$	$2.14E+03$	$4.28E+03$	$6.97E+03$	$2.89E+03$	$9.62E+03$	$6.39E+03$	$4.32E+03$	$4.15E+03$	$7.39E+03$	$2.22E+03$
	SD	$3.97E+03$	$1.26E+02$	$3.59E+03$	$3.83E+03$	$5.58E+02$	$6.52E+03$	$3.51E+03$	$1.26E+03$	$2.47E+03$	$3.66E+03$	$2.00E+02$
F22	Avg	$2.26E+03$	$2.22E+03$	$2.33E+03$	$2.26E+03$	$2.22E+03$	$2.25E+03$	$2.38E+03$	$2.25E+03$	$2.23E+03$	$2.27E+03$	$2.28E+03$
	SD	$5.42E+01$	$9.48E-01$	$7.57E+01$	$5.69E+01$	$2.38E+00$	$4.81E+01$	$9.37E+01$	$1.07E+01$	$5.87E+00$	$4.53E+01$	$7.54E+01$
F23	Avg	$2.69E+03$	$2.67E+03$	$2.69E+03$	$2.68E+03$	$2.56E+03$	$2.68E+03$	$2.77E+03$	$2.69E+03$	$2.69E+03$	$2.67E+03$	$2.59E+03$
	SD	$3.36E+01$	$1.39E-12$	$4.16E+01$	$3.30E+01$	$8.01E+01$	$3.94E+01$	$6.99E+01$	$3.10E+01$	$1.97E+01$	$3.96E+01$	$4.70E+01$
F24	Avg	$2.55E+03$	$2.52E+03$	$2.55E+03$	$2.55E+03$	$2.53E+03$	$2.57E+03$	$2.56E+03$	$2.54E+03$	$2.56E+03$	$2.55E+03$	$2.51E+03$
	SD	$2.50E+01$	$2.67E+01$	$4.14E+01$	$2.55E+01$	$2.24E+01$	$3.25E+01$	$2.04E+01$	$1.70E+01$	$3.95E+01$	$2.46E+01$	$1.78E+01$
F25	Avg	$2.70E+03$	$2.70E+03$	$2.70E+03$	$2.70E+03$	<b><math>2.65E+03</math></b>	$2.70E+03$	$2.70E+03$	$2.70E+03$	$2.70E+03$	$2.70E+03$	$2.66E+03$
	SD	$1.51E+01$	$2.62E-02$	$1.64E+01$	$1.33E+01$	$8.33E+00$	$1.23E+01$	$9.47E+00$	$7.34E+00$	$1.08E+00$	$1.11E+01$	$1.93E+01$
F26	Avg	$2.70E+03$	$2.70E+03$	$2.71E+03$	$2.70E+03$	$2.70E+03$	$2.70E+03$	$2.70E+03$	$2.70E+03$	$2.70E+03$	$2.70E+03$	$2.70E+03$
	SD	$1.57E-01$	$2.24E-02$	$4.22E+01$	$1.23E-01$	$2.22E+00$	$1.20E-01$	$6.66E-01$	$1.04E-01$	$6.24E-02$	$2.16E-01$	$1.82E+01$
F27	Avg	$3.05E+03$	$3.04E+03$	$3.07E+03$	$3.06E+03$	$2.94E+03$	$3.08E+03$	$3.12E+03$	$2.75E+03$	$2.98E+03$	$3.16E+03$	$3.07E+03$
	SD	$1.38E+02$	$9.74E+01$	$3.87E+01$	$1.22E+02$	$1.67E+02$	$1.07E+02$	$6.58E+01$	$9.85E+01$	$1.70E+02$	$4.58E+01$	$7.54E+01$
F28	Avg	$3.16E+03$	$3.12E+03$	$3.18E+03$	$3.16E+03$	<b><math>3.11E+03</math></b>	$3.16E+03$	$3.23E+03$	$3.15E+03$	$3.15E+03$	$3.20E+03$	$3.16E+03$
	SD	$3.53E+01$	$6.81E-02$	$4.11E+01$	$4.18E+01$	$7.75E+01$	$5.14E+01$	$5.45E+01$	$4.48E+00$	$3.25E+01$	$8.48E+01$	$5.15E+01$
F29	Avg	$3.28E+03$	<b><math>3.10E+03</math></b>	$2.35E+05$	$3.35E+03$	$3.17E+03$	$3.35E+03$	$1.28E+04$	$4.55E+03$	$3.27E+03$	$4.64E+03$	$3.13E+03$
	SD	$2.13E+02$	<b><math>4.30E-01</math></b>	$6.74E+05$	$4.16E+02$	$3.79E+01$	$2.70E+02$	$1.10E+04$	$8.79E+02$	$9.33E+01$	$6.22E+03$	$1.14E+01$
F30	Avg	$3.57E+03$	$3.22E+03$	$3.45E+03$	$3.58E+03$	$3.40E+03$	$3.49E+03$	$5.77E+03$	$3.84E+03$	$3.44E+03$	$3.93E+03$	$3.38E+03$
	SD	$2.07E+02$	$1.10E+00$	$2.57E+02$	$2.88E+02$	$1.14E+02$	$1.16E+02$	$2.78E+03$	$3.04E+02$	$7.40E+01$	$4.20E+02$	$9.33E+01$

**Table 9**

Wilcoxon signed-rank sum test for CEC 2014 benchmark between each algorithm against YDSE.

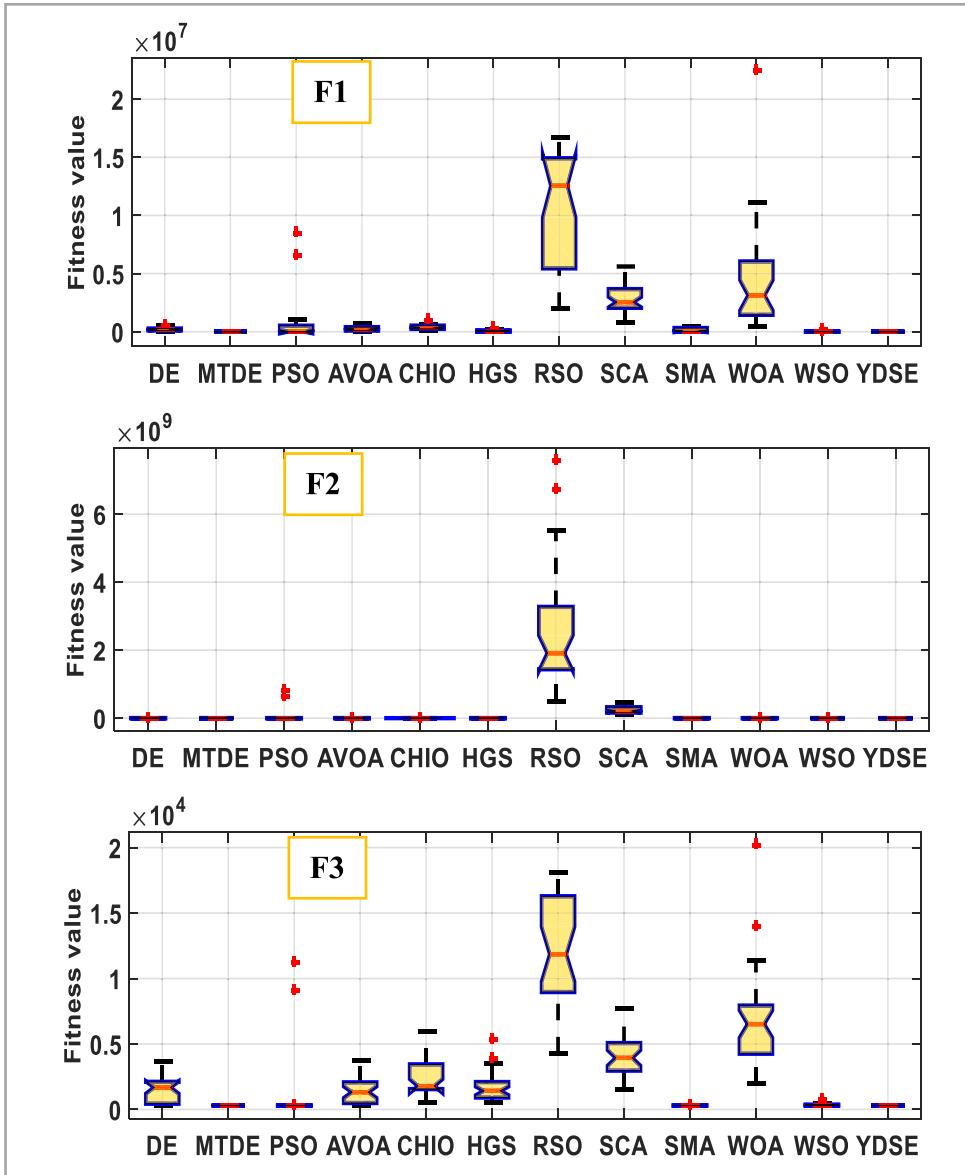
Fn.	DE	MTDE	PSO	AVOA	CHIO	HGS	RSO	SCA	SMA	WOA	WSO
F1	<b>0.00</b>	1.00	<b>0.00</b>								
F2	<b>0.00</b>	1.00	<b>0.00</b>	0.11							
F3	<b>0.00</b>	1.00	<b>0.00</b>								
F4	<b>0.00</b>	1.00	<b>0.00</b>								
F5	0.12	<b>0.00</b>	<b>0.00</b>	0.15	<b>0.00</b>	0.85	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.01</b>	<b>0.00</b>
F6	<b>0.00</b>	<b>0.00</b>	0.43	<b>0.00</b>							
F7	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	0.01	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
F8	<b>0.00</b>	<b>0.00</b>	0.90	<b>0.00</b>							
F9	<b>0.00</b>	0.06	0.19	<b>0.00</b>	0.04						
F10	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	0.01	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
F11	<b>0.00</b>	<b>0.02</b>	0.80	<b>0.00</b>	0.02	0.23	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
F12	<b>0.00</b>	<b>0.00</b>	0.90	<b>0.00</b>	<b>0.00</b>	0.73	<b>0.00</b>	<b>0.00</b>	0.73	<b>0.00</b>	<b>0.00</b>
F13	<b>0.00</b>	<b>0.00</b>	0.99	<b>0.00</b>	0.03	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.01</b>	<b>0.00</b>	0.02
F14	<b>0.00</b>	0.36	<b>0.00</b>	<b>0.00</b>							
F15	<b>0.00</b>	0.94	<b>0.00</b>								
F16	<b>0.00</b>	<b>0.00</b>	0.40	<b>0.00</b>	0.18	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.01</b>	<b>0.00</b>	0.71
F17	<b>0.00</b>	0.80	<b>0.00</b>								
F18	<b>0.00</b>										
F19	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	0.57	0.46	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	0.21
F20	<b>0.00</b>	0.03	<b>0.00</b>								
F21	<b>0.00</b>										
F22	<b>0.00</b>										
F23	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	0.57	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
F24	<b>0.00</b>	0.22									
F25	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	0.23	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
F26	<b>0.00</b>	0.01	<b>0.00</b>								
F27	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	0.63	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	0.08	<b>0.00</b>	<b>0.00</b>
F28	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	0.02	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
F29	<b>0.00</b>										
F30	<b>0.00</b>										

Tables 10 and 11 record the results of twelve algorithms for the CEC 2017 benchmark. For functions F1, F3, F4 and F9, the YDSE optimizer achieves the global optimum with a SD value of zero. YDSE outperforms its competitors on 18 test functions out of 29 and as a result, YDSE is ranked first among the algorithms. We can observe from the results that DE comes in the second rank and yields better average values for F1, F3, F6, F9 and F15. MTDE comes in the third rank with obtaining the minimum average values in four functions (F7, F10, F17, F19). In the same context, CHIO achieves the minimum Avg values in F20, F23, F24 and F25. The results demonstrate the robustness of the suggested algorithm.

A comparison is conducted among the algorithms using the Wilcoxon signed rank-sum test. The Wilcoxon signed rank-sum test compares how significant is the statistical difference among the results obtained by both YDSE and each algorithm. The test is performed with a significance threshold value of 0.05 for all 29 CEC 2017 functions. The *p*-values extracted from the test are displayed in Table 12. The *p*-values smaller than 0.05 are indicated by bold font. The proposed YDSE outperforms the state-of-the-art algorithm by achieving smaller *p*-values (<0.05), which might serve as stronger evidence of a meaningful difference.

Figs. 25–28 depict the boxplots of best fitness values achieved by the algorithms during thirty independent runs for the CEC 2017 benchmark. The boxplots in the figures display a visual comparison among the algorithm using four different types of the test functions, including unimodal, multimodal, hybrid and composite test functions. In the different types of the test functions, YDSE obtains the lower fitness values which can be seen from the low boxplot height. Also, YDSE has a smaller median and a narrower interquartile range across the board of examined functions. This is seen as proof of the YDSE algorithm's sturdiness and stability.

In Fig. 29, we provide a comparison among the algorithms based on the average CPU time for the CEC 2017 benchmark. PSO achieves the worst time with a value of 5.9214 s, while RSO achieves the best CPU time of 0.6753 s. Moreover, YDSE comes in the fourth rank after RSO, WOA and WSO. On the other hand, YDSE

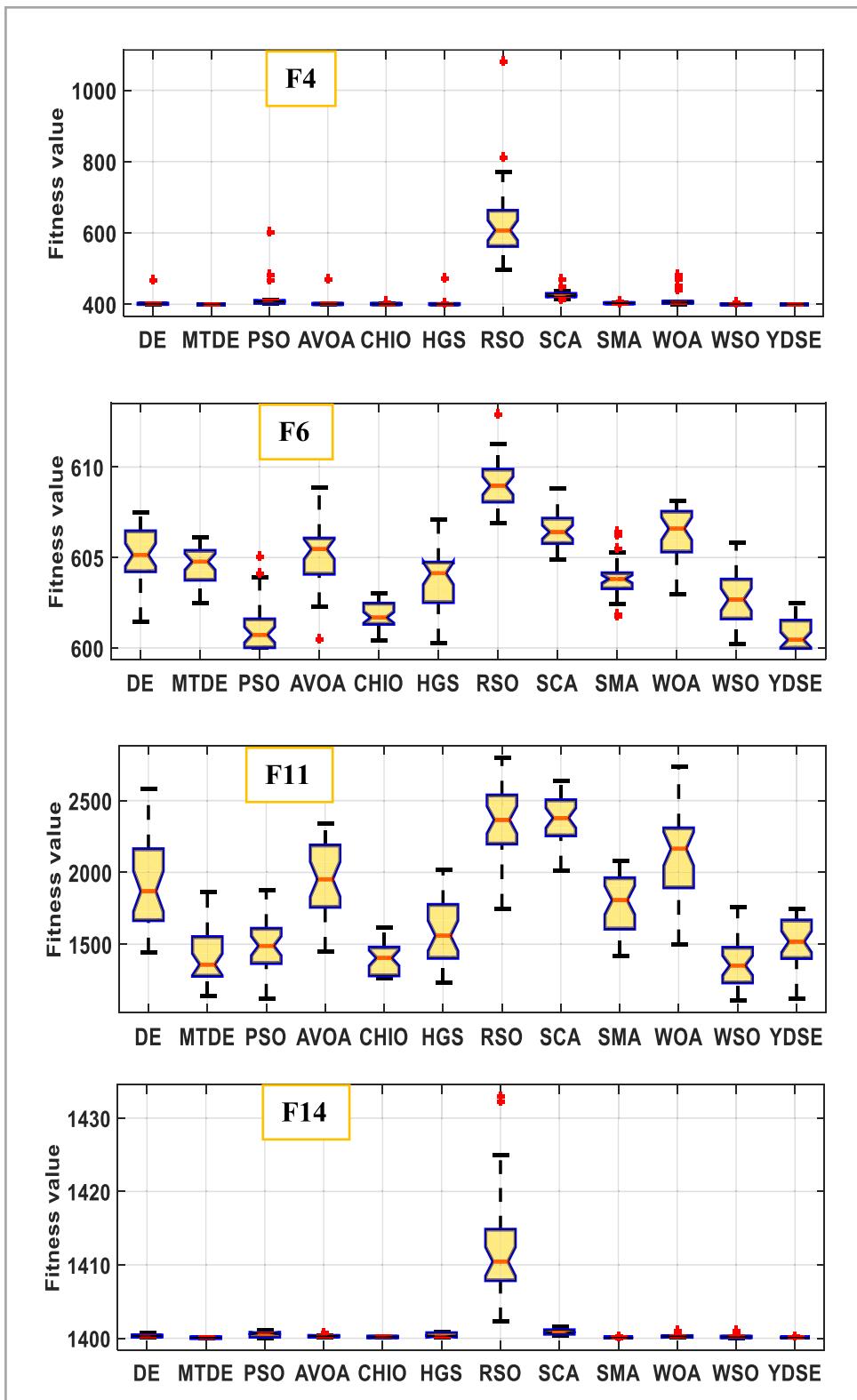


**Fig. 19.** The boxplots for the unimodal functions of CEC 2014 benchmark. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

consumes less CPU time than eight algorithms, such as DE, MTDE, PSO, AVOA, CHIO, HGS, SCA and SMA. Although MTDE and CHIO performed well in previous experiments, we discovered that they consume more CPU time (3.8709 and 3.0000 s, respectively), whereas YDSE consumes 0.8233 s.

#### 6.6. Results for CEC 2022 benchmark

In this experiment, we have employed the latest CEC 2022 benchmark which consists of twelve test functions and the specification of those functions are listed in Table 5. In this subsection, we examine the performance of the algorithms on CEC 2022 using dimensions of 10 and 20. Also, in this experiment, we adhere to the competition rules throughout the experiment. Consequently, the number of separate runs is limited to 30 run and the total number



**Fig. 20.** The boxplots for the multimodal functions of CEC 2014 benchmark. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

**Table 10**

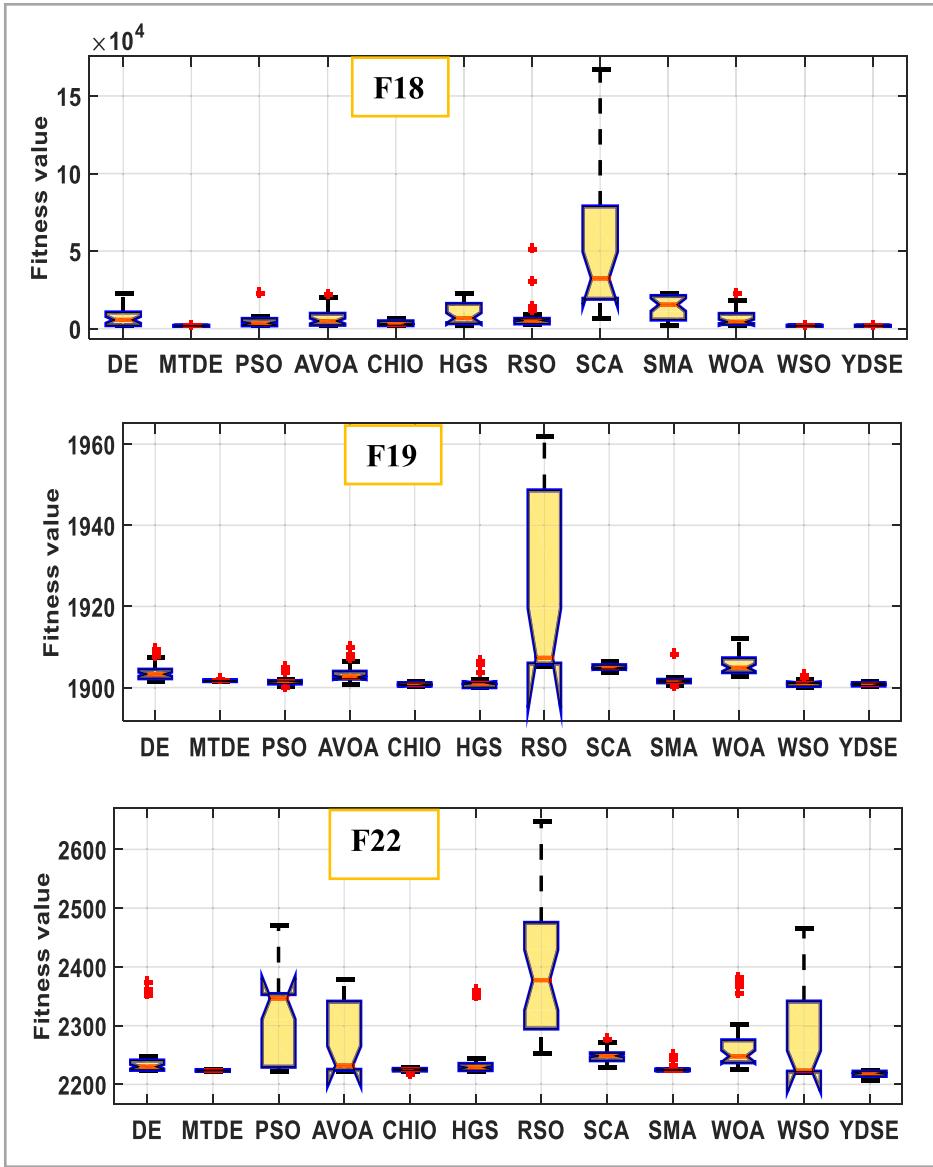
The results of the different algorithms for CEC 2017 test functions (F1–F16).

Fn.	DE	MTDE	PSO	AVOA	CHIO	HGS	RSO	SCA	SMA	WOA	WSO	YDSE	
F1	Avg	<b>1.0000E+02</b>	1.0000E+02	3.7841E+07	2.3394E+03	2.0378E+03	7.0741E+03	2.6554E+09	4.5957E+08	8.0739E+03	6.1942E+04	1.1271E+03	<b>1.0000E+02</b>
	SD	<b>0.0000E+00</b>	2.6389E-15	2.0723E+08	2.3256E+03	1.6796E+03	4.0774E+03	1.6567E+09	2.0118E+08	4.2559E+03	1.5319E+05	3.3142E+03	<b>0.0000E+00</b>
F3	Avg	<b>3.0000E+02</b>	3.0000E+02	3.7228E+02	3.0000E+02	4.2195E+03	3.0000E+02	4.5286E+03	9.2601E+02	3.0000E+02	4.1855E+02	3.2717E+02	<b>3.0000E+02</b>
	SD	<b>0.0000E+00</b>	1.4928E-14	3.9592E+02	1.9175E-13	2.3814E+03	3.3185E-11	2.5506E+03	4.2801E+02	8.3632E-05	1.0910E+02	1.4880E+02	<b>0.0000E+00</b>
F4	Avg	4.0062E+02	4.0000E+02	4.2129E+02	4.0088E+02	4.0095E+02	4.0475E+02	6.1123E+02	4.2907E+02	4.0295E+02	4.1626E+02	4.0532E+02	<b>4.0000E+02</b>
	SD	2.8883E-01	3.3380E-14	4.7619E+01	5.7370E-01	1.1340E+00	1.6057E+01	1.4141E+02	1.0592E+01	8.9117E-01	3.0760E+01	2.1241E+01	<b>0.0000E+00</b>
F5	Avg	5.0896E+02	5.0710E+02	5.1052E+02	5.2912E+02	5.0851E+02	5.1670E+02	5.6681E+02	5.3945E+02	5.1206E+02	5.4989E+02	5.3716E+02	<b>5.0621E+02</b>
	SD	1.9602E+00	2.4474E+00	5.3257E+00	1.2017E+01	2.6341E+00	7.3419E+00	1.1100E+01	7.2095E+00	6.0860E+00	1.8464E+01	1.5913E+01	<b>2.4329E+00</b>
F6	Avg	<b>6.0000E+02</b>	6.0000E+02	6.0019E+02	6.0517E+02	6.0000E+02	6.0048E+02	6.4083E+02	6.1481E+02	6.0003E+02	6.2872E+02	6.2191E+02	6.0000E+02
	SD	<b>0.0000E+00</b>	2.9856E-14	5.4434E-01	5.2977E+00	4.0124E-03	7.2076E-01	6.4262E+00	3.5725E+00	1.5138E-02	1.2640E+01	1.0048E+01	2.6295E-03
F7	Avg	7.1980E+02	<b>7.1504E+02</b>	7.1849E+02	7.5072E+02	7.1914E+02	7.2601E+02	7.9592E+02	7.6542E+02	7.2085E+02	7.7508E+02	7.7000E+02	7.1570E+02
	SD	1.9966E+00	5.2072E+00	3.1626E+00	1.1700E+01	4.0291E+00	1.0557E+01	1.3001E+01	5.7062E+00	3.9800E+00	1.8890E+01	2.4894E+01	2.1072E+00
F8	Avg	8.0888E+02	8.0939E+02	8.0793E+02	8.2736E+02	8.0970E+02	8.1699E+02	8.3753E+02	8.3197E+02	8.1187E+02	8.3553E+02	8.2925E+02	<b>8.0557E+02</b>
	SD	2.2939E+00	3.9253E+00	3.8882E+00	9.9074E+00	2.5419E+00	6.4080E+00	9.7835E+00	5.5340E+00	5.6702E+00	1.0002E+01	1.3224E+01	<b>2.3885E+00</b>
F9	Avg	<b>9.0000E+02</b>	9.0000E+02	<b>9.0000E+02</b>	9.6734E+02	9.0015E+02	9.0000E+02	1.2390E+03	9.6113E+02	9.0000E+02	1.2217E+03	1.1720E+03	<b>9.0000E+02</b>
	SD	<b>0.0000E+00</b>	2.1111E-14	<b>0.0000E+00</b>	1.0544E+02	2.0694E-01	1.1198E-10	1.3739E+02	2.6374E+01	8.2972E-05	2.8458E+02	1.7263E+02	<b>0.0000E+00</b>
F10	Avg	1.2482E+03	<b>1.1492E+03</b>	1.3490E+03	1.6633E+03	1.2743E+03	1.4291E+03	2.3133E+03	2.0598E+03	1.5836E+03	1.9979E+03	2.1078E+03	1.3402E+03
	SD	1.6346E+02	<b>1.2552E+02</b>	2.1437E+02	2.3647E+02	9.7438E+01	1.9669E+02	1.8920E+02	1.8056E+02	2.2840E+02	3.4774E+02	4.2438E+02	1.2708E+02
F11	Avg	1.1002E+03	1.1013E+03	1.1095E+03	1.1235E+03	1.1065E+03	1.1125E+03	1.4179E+03	1.1654E+03	1.1077E+03	1.1733E+03	1.1633E+03	<b>1.1001E+03</b>
	SD	3.5780E-01	6.8801E-01	1.8027E+01	1.0279E+01	2.7650E+00	1.0640E+01	3.8282E+02	1.6553E+01	3.2142E+00	5.8736E+01	3.9982E+01	<b>2.5243E-01</b>
F12	Avg	1.2744E+03	1.2380E+03	4.5656E+05	2.8372E+04	1.2192E+05	1.7220E+04	2.2202E+07	5.4215E+06	4.3614E+04	3.5650E+06	1.8499E+07	<b>1.2044E+03</b>
	SD	6.2724E+01	6.3765E+01	1.7172E+06	2.0404E+04	1.5522E+05	1.4206E+04	6.2995E+07	4.1979E+06	1.7520E+04	3.9053E+06	1.0131E+08	<b>2.1636E+01</b>
F13	Avg	1.3046E+03	1.3049E+03	6.4465E+03	1.1382E+04	3.7533E+03	9.2937E+03	2.8803E+04	1.5033E+04	1.1388E+04	1.7113E+04	1.8075E+03	<b>1.3018E+03</b>
	SD	1.3902E+00	3.0951E+00	6.8952E+03	9.7689E+03	4.6754E+03	1.0558E+04	1.7855E+04	9.9899E+03	1.1472E+04	1.2176E+04	6.0777E+02	<b>2.0269E+00</b>
F14	Avg	1.4002E+03	1.4015E+03	1.4376E+03	1.4869E+03	1.6505E+03	1.6712E+03	2.6155E+03	1.5226E+03	1.4320E+03	1.5110E+03	1.4968E+03	<b>1.4011E+03</b>
	SD	4.0479E-01	1.0642E+00	2.3627E+01	2.7943E+01	2.2292E+02	7.8878E+02	1.5428E+03	3.4029E+01	1.1020E+01	3.1946E+01	4.9029E+01	1.1338E+00
F15	Avg	<b>1.5001E+03</b>	1.5004E+03	1.5477E+03	1.6874E+03	1.9385E+03	1.9506E+03	7.9848E+03	1.7693E+03	1.5099E+03	2.7875E+03	2.4172E+03	1.5002E+03
	SD	2.4322E-01	5.7017E-01	6.2955E+01	4.1951E+02	4.4044E+02	4.9371E+02	5.8996E+03	1.3868E+02	5.5368E+00	1.2597E+03	2.0179E+03	1.7519E-01
F16	Avg	1.6076E+03	1.6044E+03	1.7056E+03	1.7298E+03	1.6241E+03	1.7428E+03	1.8272E+03	1.6691E+03	1.6829E+03	1.8025E+03	1.8632E+03	<b>1.6013E+03</b>
	SD	2.4423E+01	2.1585E+01	1.3234E+02	1.1303E+02	4.0333E+01	9.9046E+01	1.3439E+02	2.7194E+01	8.1953E+01	1.2470E+02	1.9136E+02	<b>5.8299E-01</b>

**Table 11**

The results of the different algorithms for CEC 2017 test functions (F17–F30).

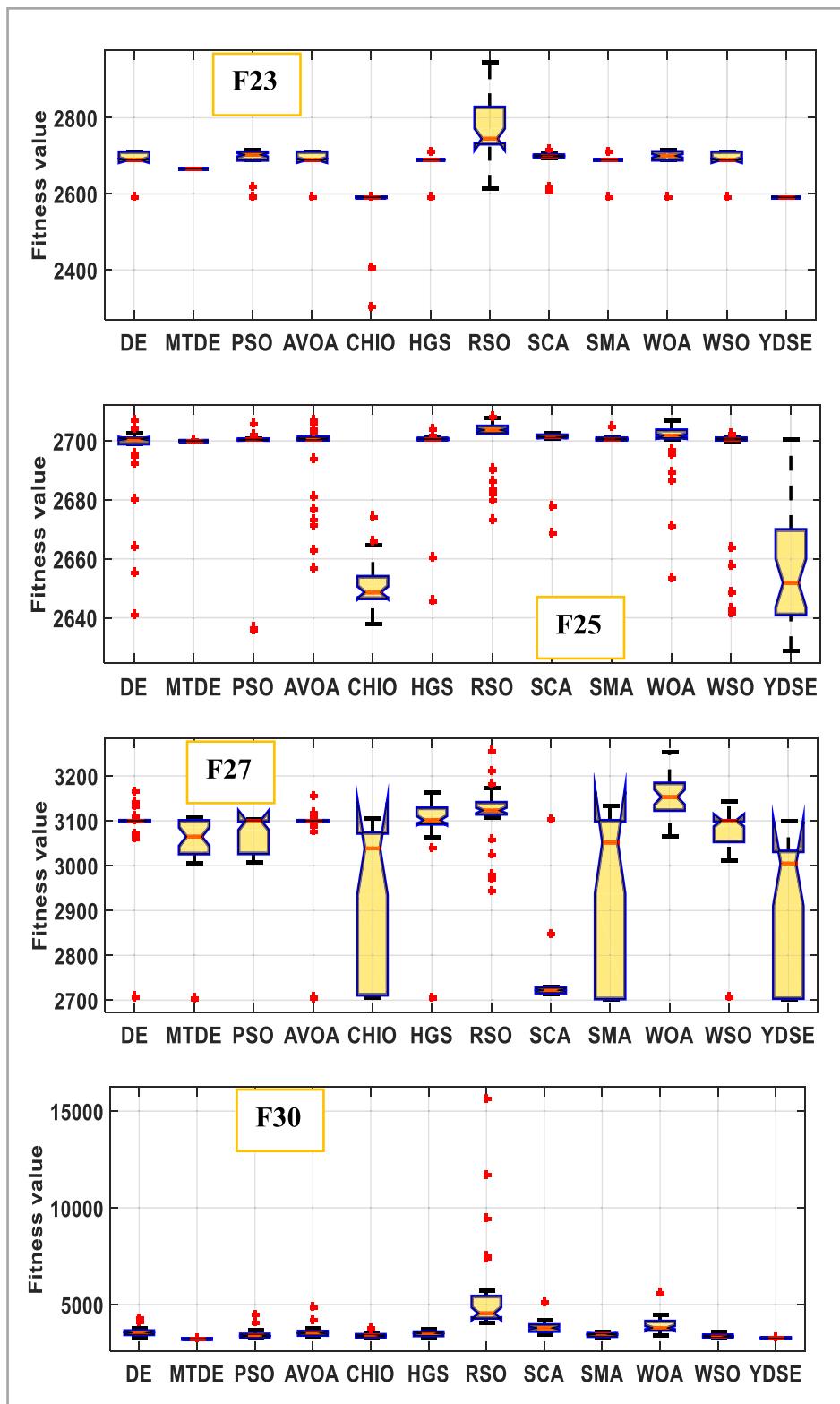
Fn.	DE	MTDE	PSO	AVOA	CHIO	HGS	RSO	SCA	SMA	WOA	WSO	YDSE	
F17	Avg	1782.72E+03	<b>1.7009E+03</b>	1.7492E+03	1.7564E+03	1.7047E+03	1.7605E+03	1.7803E+03	1.7576E+03	1.7391E+03	1.7868E+03	1.8031E+03	1.7218E+03
	SD	4.7039E+01	<b>6.5612E-01</b>	4.3609E+01	2.8959E+01	2.6204E+00	5.6294E+01	1.6541E+01	9.5570E+00	2.7903E+01	3.3328E+01	6.6080E+01	6.5642E+00
F18	Avg	1.8327E+04	1.8003E+03	1.8643E+04	1.4559E+04	6.0983E+03	1.7952E+04	2.7031E+04	5.1286E+04	1.8043E+04	2.0802E+04	6.7733E+03	<b>1.8003E+03</b>
	SD	1.0965E+04	3.8420E-01	1.7896E+04	9.3025E+03	3.8739E+03	1.1794E+04	1.4696E+04	3.1285E+04	1.1327E+04	1.1909E+04	1.2787E+04	<b>3.0637E-01</b>
F19	Avg	1.8450E+04	<b>1.9000E+03</b>	6.0771E+03	3.5029E+03	2.2736E+03	5.5300E+03	2.6503E+04	2.9016E+03	1.9080E+03	1.8443E+04	5.0613E+03	1.9004E+03
	SD	2.4198E+04	<b>2.5473E-02</b>	1.0714E+04	3.2316E+03	4.3032E+02	6.0918E+03	2.8248E+04	2.8173E+03	1.8124E+01	2.2223E+04	9.4810E+03	3.5469E-01
F20	Avg	2.1162E+03	2.0003E+03	2.0685E+03	2.0562E+03	<b>2.0011E+03</b>	2.0080E+03	2.1275E+03	2.0743E+03	2.0201E+03	2.1352E+03	2.1433E+03	2.0115E+03
	SD	7.4152E+01	3.5072E-01	6.3455E+01	4.7683E+01	<b>1.0029E+00</b>	9.6518E+00	4.6817E+01	1.7607E+01	8.5329E+00	7.1167E+01	7.2825E+01	8.9695E+00
F21	Avg	2.3139E+03	2.2705E+03	2.3059E+03	2.2345E+03	2.2092E+03	2.3153E+03	2.2268E+03	2.2217E+03	2.2884E+03	2.3131E+03	2.3333E+03	<b>2.2000E+03</b>
	SD	6.4759E+01	6.0595E+01	2.8290E+01	5.7369E+01	5.2082E+00	3.2628E+01	1.2761E+01	4.1560E+01	5.2924E+01	6.9750E+01	3.9910E+01	<b>4.0498E-13</b>
F22	Avg	2.3035E+03	2.2700E+03	2.3399E+03	2.3063E+03	2.2715E+03	2.3902E+03	2.6573E+03	2.3393E+03	2.2966E+03	2.3400E+03	2.5179E+03	<b>2.2663E+03</b>
	SD	2.2589E+01	4.4450E+01	1.0553E+02	8.6098E+00	3.1157E+01	2.3711E+02	1.0663E+02	2.5484E+01	2.3827E+01	1.7637E+02	2.6894E+02	4.3502E+01
F23	Avg	2.6510E+03	2.6067E+03	2.6256E+03	2.6357E+03	<b>2.6037E+03</b>	2.6227E+03	2.6687E+03	2.6481E+03	2.6176E+03	2.6531E+03	2.6535E+03	2.6083E+03
	SD	1.9342E+01	2.4438E+00	1.5575E+01	1.5164E+01	<b>5.7022E+01</b>	8.1593E+00	1.2178E+01	6.1423E+00	6.8119E+00	1.9787E+01	2.9746E+01	3.0164E+00
F24	Avg	2.7719E+03	2.7301E+03	2.7142E+03	2.6959E+03	<b>2.5102E+03</b>	2.7599E+03	2.8182E+03	2.7533E+03	2.7448E+03	2.7625E+03	2.7809E+03	2.5788E+03
	SD	5.4621E+01	4.3574E+01	9.7832E+01	1.3099E+02	<b>2.7154E+01</b>	5.0150E+01	4.7987E+01	7.3034E+01	4.6903E+01	4.3230E+01	6.1077E+01	1.1332E+02
F25	Avg	2.9417E+03	2.8981E+03	2.9276E+03	2.9342E+03	<b>2.8309E+03</b>	2.9290E+03	3.0795E+03	2.9433E+03	2.9251E+03	2.9281E+03	2.9289E+03	2.8861E+03
	SD	2.1117E+01	9.0123E-02	2.9115E+01	2.1247E+01	1.1169E+02	2.5596E+01	8.2582E+01	1.6366E+01	2.6117E+01	6.5175E+01	2.9780E+01	<b>5.7607E+01</b>
F26	Avg	3.4341E+03	2.9000E+03	3.0832E+03	3.0259E+03	<b>2.7572E+03</b>	3.3149E+03	3.3633E+03	3.0375E+03	3.0294E+03	3.3139E+03	3.4048E+03	2.7833E+03
	SD	5.6228E+02	8.4444E-14	3.0688E+02	1.9809E+02	1.1076E+02	4.7341E+02	2.3375E+02	2.1604E+01	2.9928E+02	5.6556E+02	5.8365E+02	1.2058E+02
F27	Avg	3.1274E+03	3.0894E+03	3.1104E+03	3.0960E+03	3.0951E+03	3.0961E+03	3.1309E+03	3.1009E+03	3.0900E+03	3.1207E+03	3.1448E+03	<b>3.0890E+03</b>
	SD	3.1922E+01	1.5318E+00	1.8600E+01	6.4265E+00	2.6823E+00	1.4247E+01	4.0372E+01	1.5530E+00	9.2609E-01	2.9707E+01	3.4047E+01	<b>5.6423E-01</b>
F28	Avg	3.4033E+03	3.1000E+03	3.3618E+03	3.2595E+03	3.1062E+03	3.2227E+03	3.3253E+03	3.2256E+03	3.1559E+03	3.4140E+03	3.3810E+03	<b>3.0932E+03</b>
	SD	1.7591E+02	1.6889E-13	8.4193E+01	1.7971E+02	8.5316E+01	1.8480E+02	1.7041E+02	1.9536E+01	1.1838E+02	1.5340E+02	2.2203E+02	<b>5.8116E+01</b>
F29	Avg	3.3213E+03	3.1334E+03	3.1979E+03	3.2365E+03	3.1631E+03	3.2067E+03	3.3184E+03	3.1943E+03	3.1679E+03	3.3247E+03	3.2686E+03	<b>3.1357E+03</b>
	SD	6.8433E+02	3.7983E+00	5.7529E+01	5.8406E+01	2.6434E+01	4.9884E+01	7.3528E+01	2.1559E+01	4.8568E+01	1.0537E+02	7.0051E+01	<b>1.8692E+01</b>
F30	Avg	4.2781E+05	3.4429E+03	4.2269E+05	3.8261E+04	5.1124E+04	2.9839E+05	3.3819E+06	5.0716E+05	3.4445E+03	3.6426E+05	3.6425E+06	<b>3.3970E+03</b>
	SD	5.3377E+05	4.7708E+01	6.5090E+05	1.4791E+05	6.5465E+04	4.1217E+05	5.6046E+06	4.8434E+05	7.7347E+01	4.4475E+05	7.0412E+06	<b>9.2097E+00</b>



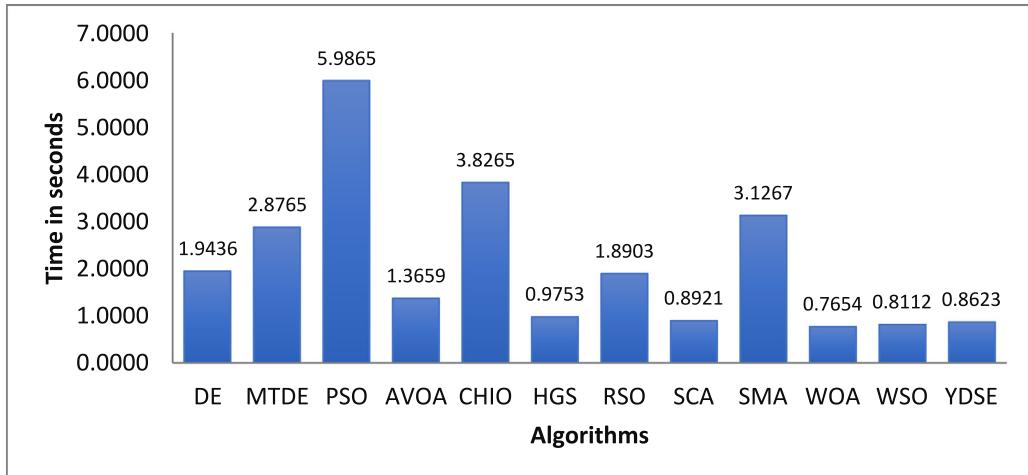
**Fig. 21.** The boxplots for the hybrid functions of CEC 2014 benchmark. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

of evaluations has been restricted to 200,000 for all the algorithms. Moreover, the additional configurations are figured up according to Table 6. Fig. 30 displays YDSE's convergence curve for several CEC 2022 benchmark test functions (F2, F5, F7, F9, F10, F12) using a dimension of 20. Moreover, the figure depicts the link between the mean fitness values obtained from thirty consecutive runs and the number of iterations. We can see that YDSE early converges towards the optimal solution during the first iterations.

The Avg and SD are recorded in Table 13, which are utilized as two performance measures of the different algorithms. The table presents the results of the CEC 2022 for dimension of 10. By inspecting the results, we find that YDSE gets the global optimum for F1 and F5 with a SD value of zero. The YDSE optimizer surpasses its competitors on the majority of the CEC 2022 test functions except for F3, F7, F8 and F9. DE gets better average value in F1, F3 and F8 in comparison with the other algorithms. Also, MTDE attains the minimum Avg value in F7. Hence, the outcomes demonstrate the strength of the suggested YDSE algorithm.



**Fig. 22.** The boxplots for the composite functions of CEC 2014 benchmark. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

**Fig. 23.** The average CPU time over thirty test functions of the CEC 2014 benchmark.**Table 12**

Wilcoxon signed-rank sum test for CEC 2017 between each algorithm against YDSE.

Fn.	DE	MTDE	PSO	AVOA	CHIO	HGS	RSO	SCA	SMA	WOA	WSO
F1	<b>0.00</b>	0.10	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	0.10	<b>0.00</b>	<b>0.00</b>	0.10	<b>0.00</b>	<b>0.00</b>
F3	0.65	0.32	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	0.32	0.32	0.32	0.32	0.32	<b>0.00</b>
F4	<b>0.00</b>	0.18	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	0.18	<b>0.00</b>	<b>0.00</b>
F5	<b>0.00</b>	<b>0.00</b>	<b>0.01</b>	<b>0.00</b>	0.36	<b>0.00</b>	<b>0.00</b>	<b>0.04</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
F6	<b>0.00</b>	<b>0.00</b>	<b>0.01</b>	<b>0.00</b>							
F7	<b>0.00</b>	<b>0.00</b>	0.34	<b>0.00</b>	0.36	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
F8	<b>0.00</b>	<b>0.00</b>	<b>0.02</b>	<b>0.00</b>	0.30	<b>0.00</b>	<b>0.00</b>	0.48	<b>0.00</b>	<b>0.00</b>	<b>0.02</b>
F9	<b>0.00</b>	<b>0.00</b>	0.64	<b>0.00</b>	0.08						
F10	<b>0.00</b>	<b>0.00</b>	0.16	<b>0.00</b>	0.73	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.02</b>
F11	<b>0.00</b>	<b>0.00</b>	0.64	<b>0.00</b>	0.52	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
F12	<b>0.00</b>										
F13	<b>0.00</b>										
F14	<b>0.00</b>	<b>0.00</b>	0.11	<b>0.00</b>	<b>0.02</b>	<b>0.00</b>	0.18	0.78	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
F15	<b>0.00</b>	<b>0.00</b>	<b>0.02</b>	0.52	0.53	<b>0.00</b>	0.57	<b>0.01</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
F16	<b>0.00</b>	<b>0.00</b>	0.14	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.01</b>	<b>0.00</b>	<b>0.00</b>	0.40
F17	<b>0.00</b>	<b>0.00</b>	0.40	<b>0.00</b>	<b>0.00</b>	0.12	<b>0.02</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	0.14
F18	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.04</b>	<b>0.00</b>						
F19	<b>0.05</b>	<b>0.00</b>	<b>0.00</b>	<b>0.02</b>	0.01	<b>0.00</b>	<b>0.01</b>	<b>0.02</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
F20	<b>0.00</b>	<b>0.00</b>	0.93	<b>0.00</b>	<b>0.00</b>	<b>0.04</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.36</b>
F21	<b>0.00</b>	<b>0.00</b>	0.35	<b>0.00</b>	<b>0.00</b>	0.38	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
F22	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.01</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
F23	<b>0.00</b>	<b>0.00</b>	0.88	<b>0.00</b>	0.50	0.74	<b>0.00</b>	<b>0.02</b>	<b>0.00</b>	<b>0.00</b>	<b>0.03</b>
F24	<b>0.00</b>	<b>0.00</b>	<b>0.05</b>	<b>0.00</b>	<b>0.05</b>	0.30	<b>0.01</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
F25	0.74	<b>0.00</b>	0.38	<b>0.00</b>	0.06	0.07	0.98	0.56	<b>0.00</b>	0.21	<b>0.00</b>
F26	<b>0.05</b>	<b>0.00</b>	0.47	<b>0.00</b>	<b>0.02</b>	0.75	0.43	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	0.81
F27	<b>0.00</b>	<b>0.00</b>	0.11	<b>0.00</b>	<b>0.00</b>	0.11	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.05</b>
F28	0.60	<b>0.00</b>	0.52	<b>0.00</b>	<b>0.00</b>	0.80	<b>0.00</b>	<b>0.02</b>	<b>0.00</b>	<b>0.00</b>	0.17
F29	<b>0.00</b>	<b>0.00</b>	<b>0.02</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.04</b>	<b>0.00</b>	<b>0.00</b>	<b>0.02</b>
F30	0.05	<b>0.00</b>	0.08	<b>0.00</b>	0.10	<b>0.00</b>	<b>0.04</b>	<b>0.01</b>	<b>0.00</b>	<b>0.00</b>	0.53

The Wilcoxon signed-rank test is conducted for the CEC 2022 benchmark to compare the YDSE optimizer with each algorithm using a significance threshold of 0.05, and the resulting p-values were recorded in [Table 14](#). The p-values indicate if the difference among the results of both YDSE and each algorithm is statistically significant. From the table, YDSE beats all other algorithms on the most functions with a  $p$ -value  $< 0.05$ . [Table 13](#) shows that YDSE produces the lowest average values for most functions, whereas [Table 14](#) shows that YDSE's findings

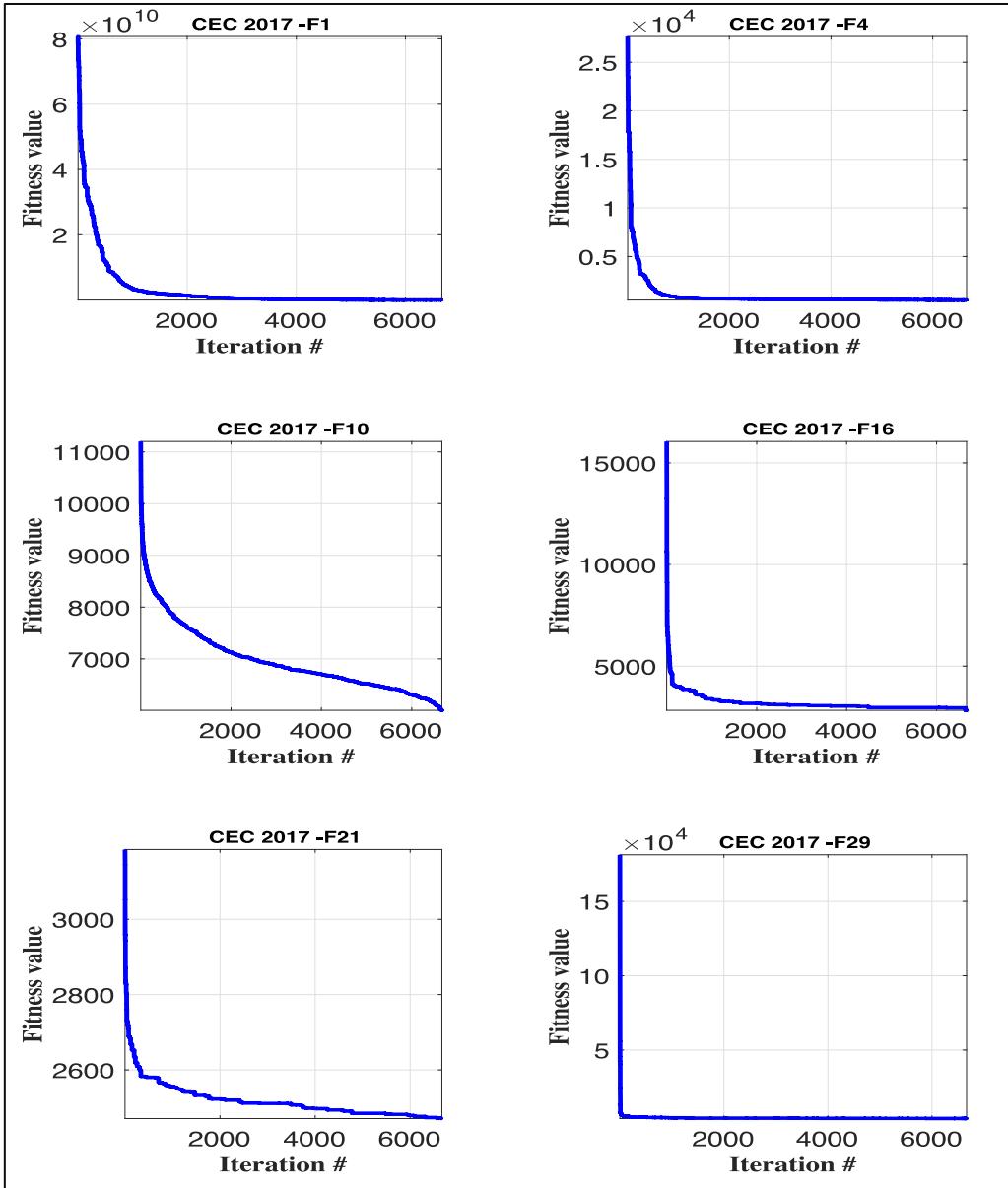


Fig. 24. The convergence curve of YDSE over some CEC 2017 test functions.

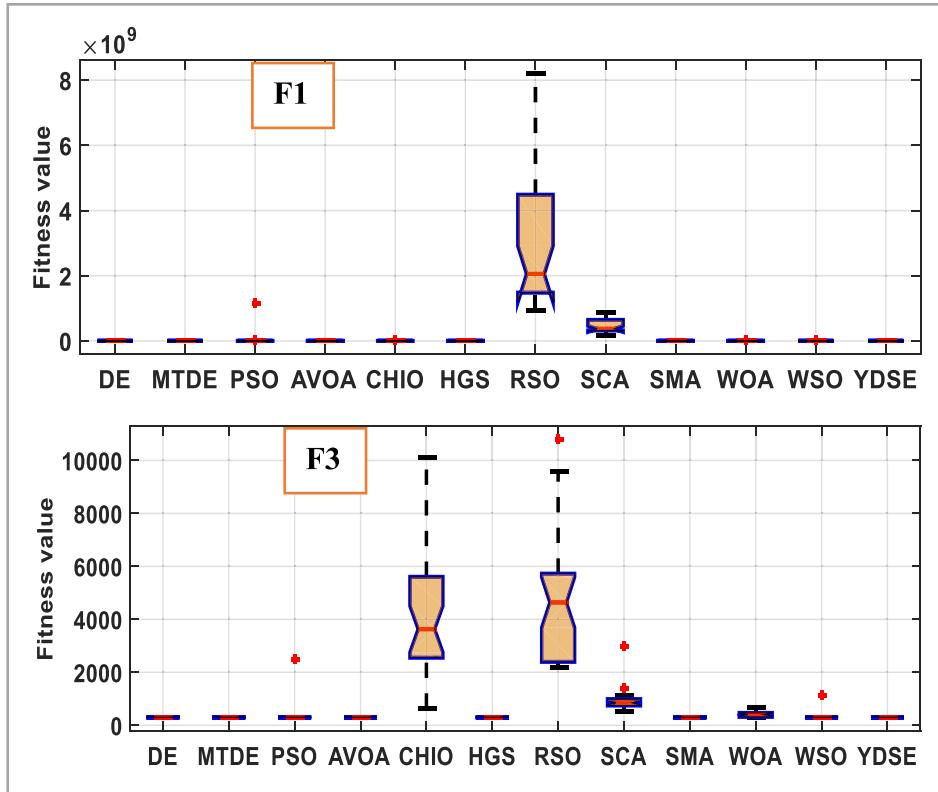
vary statistically from those obtained by its rivals. Therefore, YDSE is superior to its competitors. Fig. 31 provides the average CPU time taken by each algorithm on the CEC 2022 benchmark. WOA obtains the minimum time of 0.6932 s, while PSO obtains the worst time of 5.4821 s. YDSE can solve each function of the CEC 2022 with an average time of 0.86.

Table 15 shows the results of the algorithms on CEC 2022 with a dimension of 20. From the results, we can infer that YDSE achieves the minimum average values in seven out of twelve test functions. Moreover, DE gets better results in F3 and F5 compared to the other algorithms. Also, CHIO beats the other algorithms on F10 and F11, while MTDE obtains the best results for F1. Table 16 records the output of the Wilcoxon signed-rank test on CEC 2022 using a dimension of 20. Again, the table shows that the YDSE results significantly differ from those obtained by the other algorithms with a  $p$ -value  $< 0.05$  in most of the test functions. Finally, from Tables 15 and 16, we can conclude that YDSE keeps its outperformance over its rivals even in a higher dimension.

**Table 13**

The results of the different algorithms for CEC 2022 test functions for dimension = 10.

Fn.	DE	MTDE	PSO	AVOA	CHIO	HGS	RSO	SCA	SMA	WOA	WSO	YDSE	
F1	Avg	<b>3.0000E+02</b>	3.0000E+02	3.0000E+02	7.4230E+03	3.0000E+02	3.4778E+03	6.7162E+02	3.0000E+02	2.3327E+03	3.0000E+02	<b>3.0000E+02</b>	
	SD	<b>0.0000E+00</b>	2.5856E−14	1.8283E−14	1.5939E−13	3.3499E+03	1.4622E−09	2.1545E+03	2.3358E+02	1.1711E−05	1.1701E+03	5.8771E−14	<b>0.0000E+00</b>
F2	Avg	4.0709E+02	4.0372E+02	4.2355E+02	4.0072E+02	4.0080E+02	4.0570E+02	7.0636E+02	4.4299E+02	4.0307E+02	4.3026E+02	4.2574E+02	<b>4.0026E+02</b>
	SD	2.0905E+00	1.0114E+00	3.0336E+01	1.6046E+00	1.6219E+00	2.6843E+00	2.2979E+02	1.7287E+01	7.7050E−01	3.9200E+01	8.0722E+01	<b>3.0452E−01</b>
F3	Avg	<b>6.0000E+02</b>	6.0000E+02	6.0036E+02	6.0387E+02	6.0000E+02	6.0037E+02	6.3855E+02	6.1432E+02	6.0002E+02	6.1786E+02	6.2625E+02	6.0001E+02
	SD	<b>0.0000E+00</b>	2.9856E−14	6.1406E−01	5.1987E+00	7.9151E−03	8.3841E−01	6.8446E+00	3.2248E+00	1.2932E−02	8.9972E+00	9.2318E+00	7.4947E−03
F4	Avg	8.1679E+02	8.1131E+02	8.1146E+02	8.3002E+02	8.1167E+02	8.1551E+02	8.3598E+02	8.3449E+02	8.1127E+02	8.4623E+02	8.2562E+02	<b>8.0404E+02</b>
	SD	2.9561E+00	4.0771E+00	5.1232E+00	9.7578E+00	3.2913E+00	5.9692E+00	7.4818E+00	4.8254E+00	4.9804E+00	1.5767E+01	1.0366E+01	<b>1.5610E+00</b>
F5	Avg	9.0000E+02	9.0000E+02	9.0005E+02	1.3692E+03	9.0025E+02	9.0181E+02	1.2418E+03	9.5957E+02	9.0001E+02	2.2121E+03	1.1136E+03	<b>9.0000E+02</b>
	SD	1.0556E−13	2.1111E−14	1.1817E−01	5.6052E+02	2.8494E−01	9.3590E+00	1.1534E+02	3.1226E+01	2.2706E−02	7.2606E+02	1.2468E+02	<b>0.0000E+00</b>
F6	Avg	1.8003E+03	1.8005E+03	5.3085E+03	4.6925E+03	3.3607E+03	9.0037E+03	3.8958E+06	7.1173E+05	1.0704E+04	6.1154E+03	3.2075E+03	<b>1.8002E+03</b>
	SD	1.4281E−01	4.4333E−01	2.6514E+03	3.1126E+03	1.6547E+03	5.6498E+03	1.4466E+07	5.2379E+05	5.6928E+03	4.9693E+03	2.5760E+03	<b>1.1919E−01</b>
F7	Avg	2.0007E+03	<b>2.0001E+03</b>	2.0126E+03	2.0352E+03	2.0147E+03	2.0213E+03	2.0722E+03	2.0466E+03	2.0212E+03	2.0474E+03	2.0579E+03	2.0058E+03
	SD	3.6680E+00	<b>3.1760E−01</b>	9.7964E+00	7.1693E+00	6.4877E+00	1.0091E+01	1.7562E+01	5.6310E+00	8.3407E+00	1.6827E+01	3.2229E+01	5.7989E+00
F8	Avg	<b>2.2006E+03</b>	2.2016E+03	2.2273E+03	2.2261E+03	2.2194E+03	2.2202E+03	2.2610E+03	2.2291E+03	2.2218E+03	2.2294E+03	2.2263E+03	2.2032E+03
	SD	<b>3.8656E−01</b>	5.0316E+00	3.9305E+01	6.1501E+00	4.6907E+00	4.7532E+00	5.7249E+01	2.1518E+00	8.8538E−01	6.7689E+00	8.6928E+00	3.7917E+00
F9	Avg	2.5293E+03	2.5293E+03	2.5341E+03	2.4101E+03	2.4001E+03	2.4067E+03	2.6217E+03	2.5479E+03	<b>2.4000E+03</b>	2.4014E+03	2.5404E+03	2.5250E+03
	SD	0.0000E+00	0.0000E+00	1.2023E+01	3.0478E+01	4.1256E−01	2.5371E+01	3.6692E+01	9.9597E+00	<b>4.7375E−04</b>	5.6552E−01	3.4038E+01	2.3604E+01
F10	Avg	2.5421E+03	2.5002E+03	2.5727E+03	2.5000E+03	2.4016E+03	2.5000E+03	2.5158E+03	2.5014E+03	2.5000E+03	2.5710E+03	2.6291E+03	<b>2.4976E+03</b>
	SD	4.9345E+01	3.8080E−02	9.6839E+01	1.3898E−02	3.0767E+00	0.0000E+00	3.1381E+01	3.8786E−01	7.8229E−04	1.9562E+02	2.2314E+02	<b>1.4886E+01</b>
F11	Avg	2.6607E+03	2.6000E+03	2.7635E+03	2.6000E+03	2.6000E+03	2.6000E+03	2.9757E+03	2.7583E+03	2.6000E+03	2.6001E+03	2.7675E+03	<b>2.6000E+03</b>
	SD	8.5295E+01	2.9252E−13	1.1786E+02	1.3529E−11	3.7723E−04	2.8354E−11	2.4168E+02	7.8335E+00	2.5890E−03	4.8464E−02	2.7117E+02	<b>1.1942E−13</b>
F12	Avg	2.9544E+03	2.8634E+03	2.8710E+03	2.9492E+03	2.9289E+03	2.9543E+03	2.8844E+03	2.8677E+03	2.9492E+03	2.9544E+03	2.8976E+03	<b>2.8589E+03</b>
	SD	6.2790E−02	1.6957E+00	1.2924E+01	2.8138E+01	6.6473E+01	2.8712E−03	2.7279E+01	1.1658E+00	2.8085E+01	1.0084E−01	2.5654E+01	<b>5.9374E−01</b>



**Fig. 25.** The boxplots for the unimodal functions of CEC 2017 benchmark.

**Table 14**

Wilcoxon signed-rank sum test for CEC 2022 between each algorithm against YDSE.

Fn.	DE	MTDE	PSO	AVOA	CHIO	HGS	RSO	SCA	SMA	WOA	WSO
F1	1.00	1.00	1.00	1.00	<b>0.00</b>	1.00	<b>0.00</b>	<b>0.00</b>	1.00	<b>0.00</b>	1.00
F2	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.02</b>	0.16	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
F3	<b>0.00</b>	<b>0.00</b>	0.31	<b>0.00</b>	<b>0.00</b>	0.16	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
F4	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	0.00	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
F5	1.00	1.00	<b>0.01</b>	<b>0.00</b>	0.00	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	0.16	<b>0.00</b>	<b>0.00</b>
F6	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	0.00	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
F7	<b>0.00</b>	<b>0.00</b>	0.01	<b>0.00</b>	0.00	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
F8	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	0.00	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
F9	<b>0.00</b>	0.32	<b>0.00</b>	<b>0.00</b>	0.00	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	0.02
F10	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	0.00	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	0.03	<b>0.00</b>
F11	<b>0.00</b>	1.00	<b>0.00</b>	1.00	1.00	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	1.00	<b>0.00</b>	<b>0.00</b>
F12	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	0.00	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>

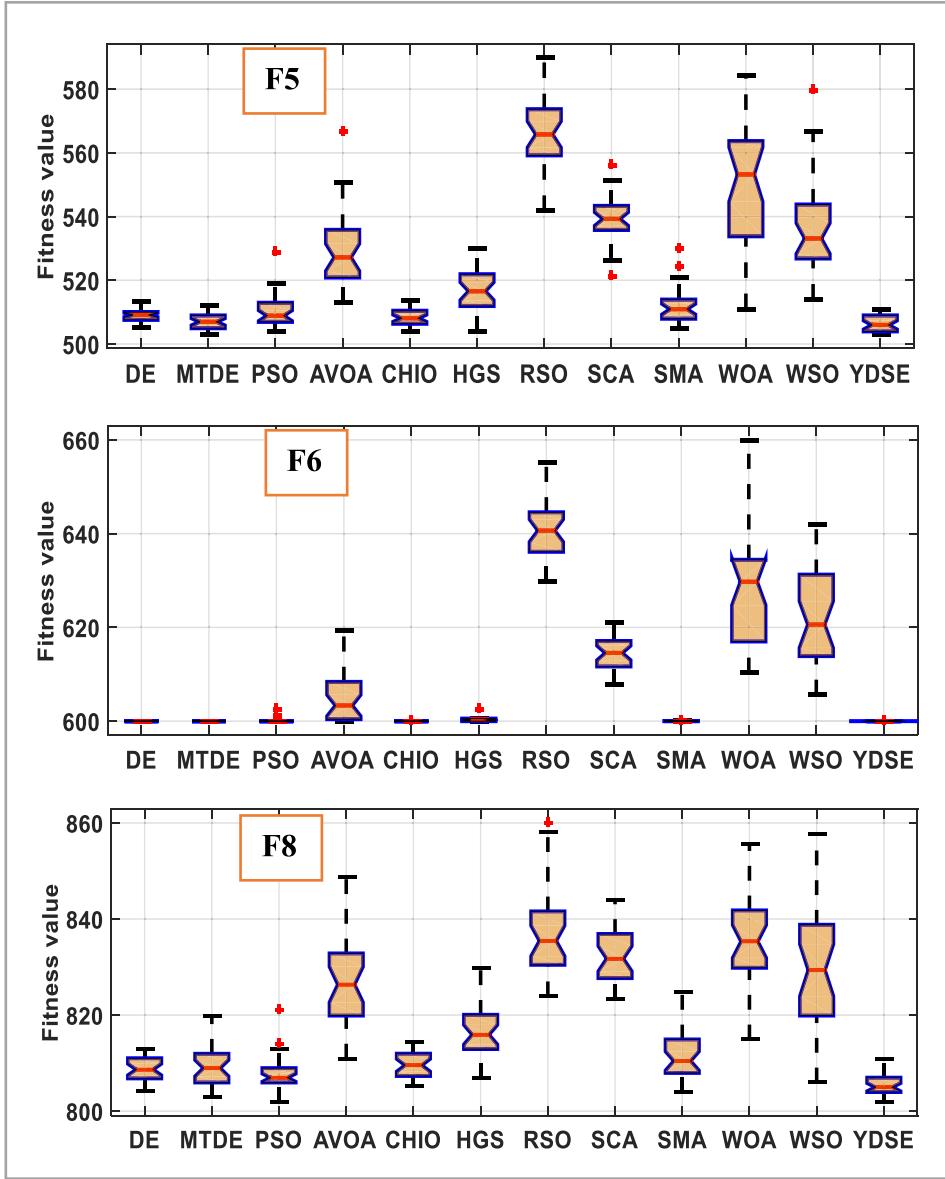
### 6.7. Comparison with the CEC winners

In order to examine the outperformance of the proposed YDSE optimizer, it is compared with other CEC winners. Two experiments are carried out using two different CEC benchmarks. For the first experiment, the YDSE is tested against the winner of the CEC 2014 competition: Linear population size reduction-Success-History Adaptation for Differential Evolution (L-SHADE) [105]. The results of the two algorithms are recorded in Table 17. The proposed YDSE outperforms L-SHADE on 21 test functions out of 30, while L-SHADE gets the minimum average values on 10 test functions. For the hybrid and composite functions (F17–F30), YDSE obtains the best outcomes except for F26. In contrast to YDSE, L-SHADE shows a bad performance for these two types of functions. For the unimodal

**Table 15**

The results of the different algorithms for CEC 2022 test functions for dimension = 20.

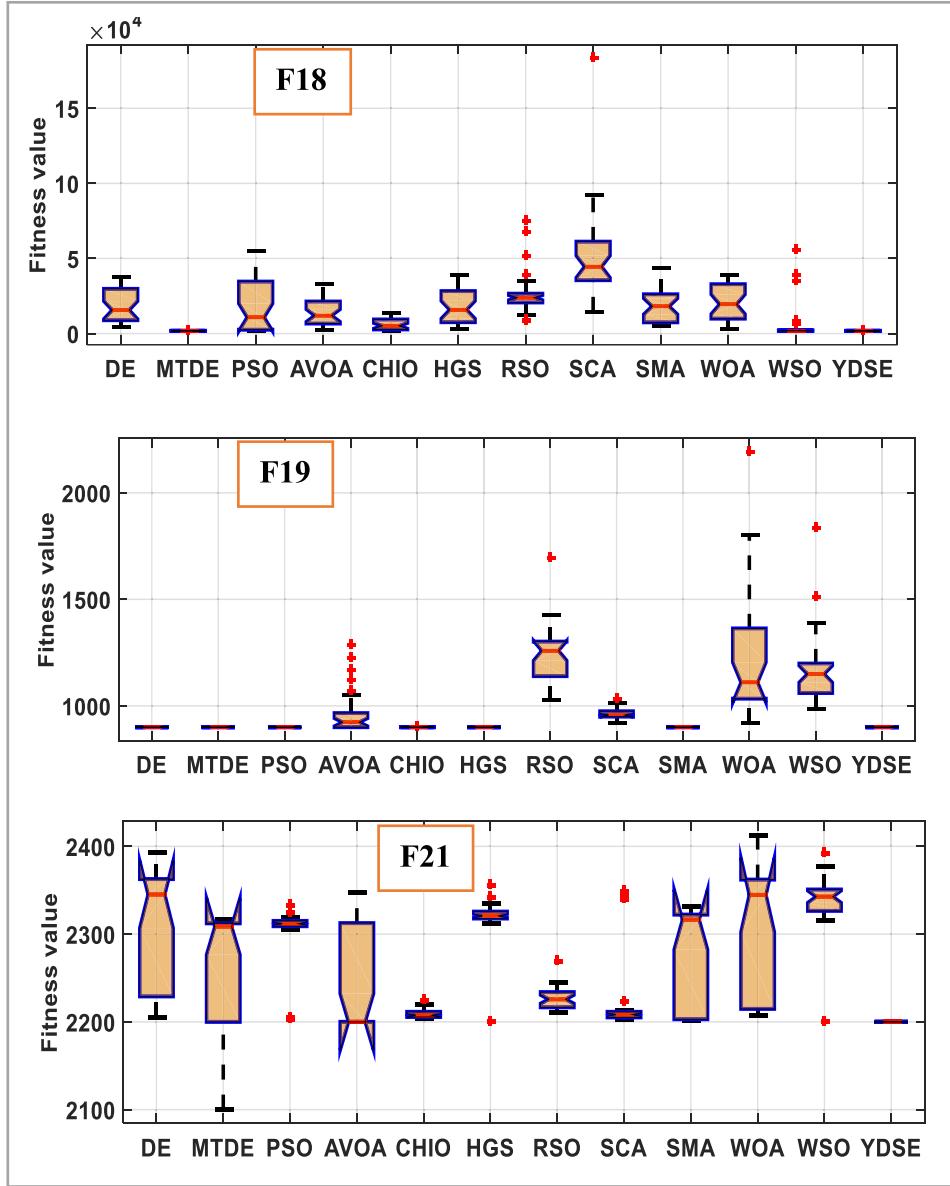
Fn.	DE	MTDE	PSO	AVOA	CHIO	HGS	RSO	SCA	SMA	WOA	WSO	YDSE	
F1	Avg	2.2157E+03	<b>3.0000E+02</b>	9.7133E+02	3.0000E+02	2.1629E+04	3.1119E+02	2.1481E+04	9.4605E+03	3.0000E+02	2.9377E+03	1.8070E+04	4.6565E+02
	SD	5.6331E+02	<b>1.5441E-13</b>	1.8466E+03	4.4253E-09	5.9773E+03	4.7115E+01	5.4331E+03	2.3057E+03	1.0780E-03	2.2875E+03	8.0477E+03	2.2659E+02
F2	Avg	4.4894E+02	4.4542E+02	4.6875E+02	4.5276E+02	4.3858E+02	4.4515E+02	1.1653E+03	6.3002E+02	4.4305E+02	4.7864E+02	7.9813E+02	<b>4.3105E+02</b>
	SD	7.6481E-01	1.2297E+01	4.5147E+01	1.4168E+01	1.8742E+01	1.8384E+01	2.7588E+02	3.9581E+01	1.5667E+01	2.9817E+01	2.6634E+02	1.6536E+01
F3	Avg	<b>6.0000E+02</b>	6.0000E+02	6.0113E+02	6.2558E+02	6.0001E+02	6.0052E+02	6.7834E+02	6.3639E+02	6.0031E+02	6.6134E+02	6.5142E+02	6.1351E+02
	SD	<b>2.1111E-14</b>	2.4410E-03	1.3497E+00	1.0107E+01	1.2083E-02	7.2287E-01	7.5931E+00	5.4005E+00	2.4906E-01	8.6531E+00	1.1039E+01	5.6394E+00
F4	Avg	8.9615E+02	8.6410E+02	8.3735E+02	8.9256E+02	8.9186E+02	9.0048E+02	9.3082E+02	9.2582E+02	8.6232E+02	9.1477E+02	9.2003E+02	<b>8.3372E+02</b>
	SD	6.7390E+00	1.4552E+01	1.0451E+01	1.9141E+01	1.7265E+01	3.2206E+01	1.5373E+01	1.0580E+01	2.1013E+01	3.5060E+01	2.8880E+01	1.2249E+01
F5	Avg	<b>9.0000E+02</b>	9.0113E+02	9.1345E+02	2.3737E+03	3.2489E+03	2.7041E+03	2.7111E+03	1.9086E+03	1.0579E+03	3.3646E+03	2.3893E+03	1.3711E+03
	SD	<b>8.7043E-14</b>	1.2122E+00	2.3162E+01	3.0984E+02	5.9871E+02	6.5500E+02	2.5969E+02	3.3654E+02	1.7090E+02	9.9920E+02	5.6204E+02	3.3655E+02
F6	Avg	7.0230E+05	3.4510E+03	5.6862E+04	7.1009E+03	1.2765E+04	1.3018E+04	2.0355E+08	7.1291E+07	2.1140E+04	7.2332E+03	3.1469E+06	<b>1.8551E+03</b>
	SD	3.8979E+05	2.9799E+03	2.7077E+05	7.2339E+03	7.7424E+03	8.3174E+03	2.1612E+08	4.4085E+07	5.8289E+03	6.3552E+03	1.2471E+07	<b>2.2890E+01</b>
F7	Avg	2.0350E+03	2.0739E+03	2.0341E+03	2.1134E+03	2.0333E+03	2.0731E+03	2.1764E+03	2.1151E+03	2.0474E+03	2.1769E+03	2.1896E+03	<b>2.0250E+03</b>
	SD	2.7560E+00	1.7924E+01	2.2707E+01	3.5456E+01	7.7394E+00	4.5719E+01	2.3577E+01	1.7240E+01	2.2454E+01	5.3875E+01	6.1323E+01	6.0642E+00
F8	Avg	2.2308E+03	2.2324E+03	2.2613E+03	2.2326E+03	2.2225E+03	2.2276E+03	2.3739E+03	2.2501E+03	2.2256E+03	2.2627E+03	2.3170E+03	<b>2.2209E+03</b>
	SD	1.0293E+00	2.1003E+00	6.3310E+01	8.1562E+00	5.7411E-01	2.2202E+01	9.4554E+01	8.7006E+00	7.1378E+00	4.3659E+01	7.7806E+01	<b>5.3339E-01</b>
F9	Avg	2.4808E+03	2.4808E+03	2.5040E+03	2.4808E+03	2.4808E+03	2.4809E+03	2.6766E+03	2.5556E+03	2.4808E+03	2.4910E+03	2.5187E+03	<b>2.4788E+03</b>
	SD	0.0000E+00	2.9398E-12	3.0321E+01	1.7522E-03	4.7618E-03	1.0425E-01	3.9253E+01	2.3694E+01	1.3743E-02	1.4070E+01	3.8729E+01	1.1359E+01
F10	Avg	2.5109E+03	2.5002E+03	3.0726E+03	2.8351E+03	<b>2.4085E+03</b>	2.6435E+03	3.7215E+03	2.6504E+03	2.7586E+03	4.0061E+03	5.1283E+03	2.7959E+03
	SD	5.7262E+01	4.4137E+01	4.9507E+02	4.0383E+02	<b>9.5900E+00</b>	1.0351E+02	1.4457E+03	6.2939E+02	2.2054E+02	1.0672E+03	1.1313E+03	7.1916E+02
F11	Avg	2.9000E+03	2.9333E+03	3.4793E+03	2.9879E+03	<b>2.6875E+03</b>	2.8967E+03	6.4905E+03	4.3528E+03	2.9840E+03	2.9475E+03	5.5243E+03	2.9208E+03
	SD	1.6734E-06	4.7946E+01	5.4582E+02	4.6727E+02	1.0062E+02	1.0981E+02	6.2260E+02	3.7625E+02	2.1157E+02	1.5635E+02	9.5143E+02	6.8662E+01
F12	Avg	2.9391E+03	2.9402E+03	2.9969E+03	2.9748E+03	2.9457E+03	2.9585E+03	3.2538E+03	3.0210E+03	2.9434E+03	3.0171E+03	3.1688E+03	<b>2.9362E+03</b>
	SD	3.3422E+00	4.8216E+00	4.2527E+01	2.0387E+01	3.9860E+00	2.0599E+01	1.5981E+02	2.0170E+01	4.6447E+00	3.3763E+01	1.3920E+02	<b>2.1253E+00</b>



**Fig. 26.** The boxplots for the multimodal functions of CEC 2017 benchmark.

functions (F1, F3), YDSE has the best results compared to L-SHADE. For the multimodal functions (F4–F16), L-SHADE achieves better results than YDSE optimizer except for F7, F9, F13, and F14.

In the second experiment, we compare the outcomes of YDSE with those obtained by two winners of the CEC 2017 competition: LSHADE-cnEpSin [106] and LSHADE-SPACMA [107]. Table 18 presents the results of the algorithms that show that YDSE can be a strong rival. The proposed YDSE comes in the first rank with getting the minimum average values on 16 test functions out of 29. LSHADE-SPACMA comes in the second rank with obtaining the minimum average values on 12 test functions, while LSHADE-cnEpSin comes in the last rank with 11 test functions. By observing the results, we can see that the performance of LSHADE-SPACMA is the best on the unimodal and multimodal test functions except for F5. Unfortunately, the performance of this algorithm begins to deteriorate and becomes the worst one for the complex functions like composite functions. Also, the LSHADE-cnEpSin achieves poor results for the hybrid functions (F10–F19), as well as the composite ones (F20–F30) except



**Fig. 27.** The boxplots for the hybrid functions of CEC 2017 benchmark.

for F11, F14, F19, F23, and F29. For our proposed algorithm, its performance is balanced for the different types of functions: unimodal, multimodal, hybrid, and composite.

#### 6.8. YDSE optimizer for constrained engineering problems

This subsection investigates the performance of YDSE Optimizer using ten well-known Engineering Optimization Problems (EOPs). Most of the EOPs seeks to maximize or minimize an objective function subject to given constraints, such as safety considerations, limited resources, and design specifications. These optimal solutions must respect the constraints imposed by the problem. The MH algorithms cannot be directly used for addressing the constrained EOPs, as they need additional techniques to handle such problems. Hence, we adopt a constraint-handling technique developed by Bayzidi et al. [108]. This technique makes it possible for any optimization method

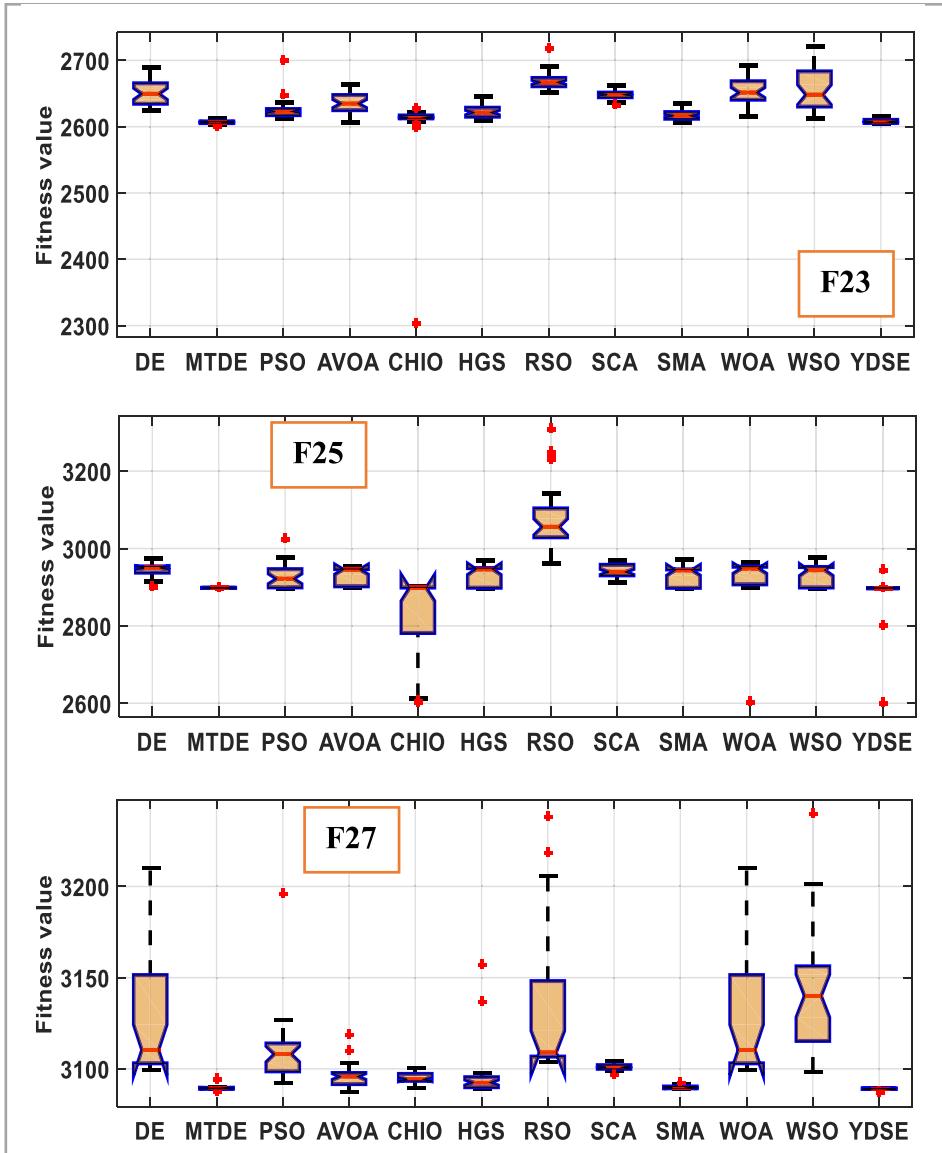
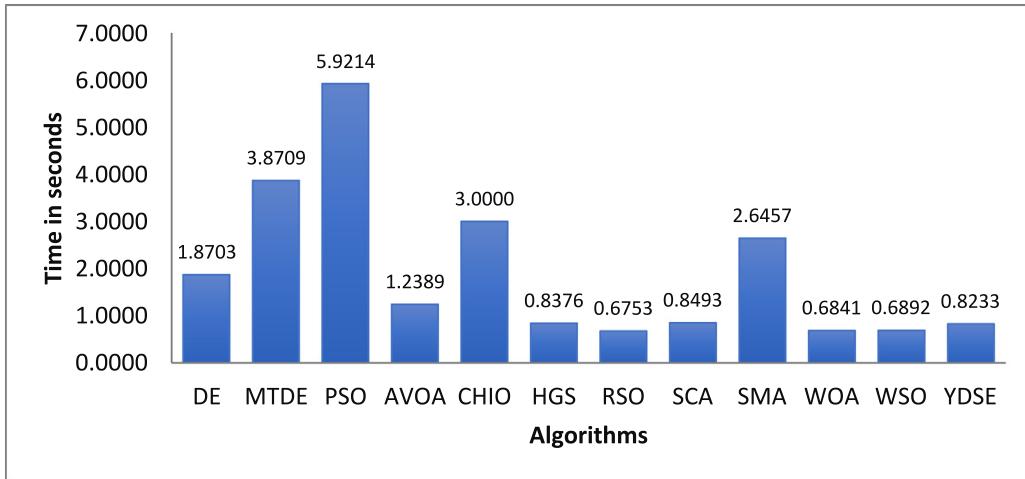


Fig. 28. The boxplots for the composite functions of CEC 2017 benchmark.

to simultaneously deal with the objectives as well as the constraints. Here are a few engineering problems to consider:

- Three-bar truss design problem.
- Tension/compression spring design problem.
- Cantilever beam design problem.
- Pressure vessel design problem.
- Speed reducer design problem.
- Gear train design problem.
- Tabular column design problem.
- I-beam vertical deflection design problem.
- Welded beam design problem.
- Piston lever design problem.



**Fig. 29.** The average CPU time over thirty test functions of the CEC 2017 benchmark.

**Table 16**

Wilcoxon signed-rank sum test for CEC 2022 between each algorithm against YDSE for dimension = 20.

Fn.	DE	MTDE	PSO	AVOA	CHIO	HGS	RSO	SCA	SMA	WOA	WSO
F1	<b>0.00</b>	<b>0.00</b>	0.31	<b>0.00</b>							
F2	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	0.11	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.01</b>	<b>0.00</b>	<b>0.00</b>
F3	<b>0.00</b>										
F4	<b>0.00</b>	<b>0.00</b>	0.14	<b>0.00</b>							
F5	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	0.94	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.03</b>	<b>0.00</b>
F6	<b>0.00</b>										
F7	<b>0.00</b>										
F8	<b>0.00</b>										
F9	<b>0.05</b>	<b>0.05</b>	<b>0.00</b>	<b>0.05</b>	<b>0.05</b>	0.67	<b>0.00</b>	<b>0.00</b>	<b>0.05</b>	<b>0.00</b>	<b>0.00</b>
F10	<b>0.00</b>	<b>0.00</b>	0.02	<b>0.08</b>	<b>0.00</b>	<b>0.04</b>	<b>0.00</b>	0.25	<b>0.07</b>	<b>0.00</b>	<b>0.00</b>
F11	<b>0.00</b>	0.53	<b>0.00</b>	0.01	<b>0.00</b>	0.11	<b>0.00</b>	<b>0.00</b>	0.21	0.28	<b>0.00</b>
F12	<b>0.00</b>										

In the next sections, the suggested method will be compared to various existing algorithms. The algorithms are assessed by 30 times with 100,000 evaluations and the remaining settings mentioned in [Table 6](#).

#### 6.8.1. Three-bar truss design problem

The design of the Three-Bar Truss (TBT) is a common EOP. It consists of a three-bar truss ( $B_1$ ,  $B_2$ ,  $B_3$ ), as depicted by [Fig. 32](#). The goal is to reduce the volume of the loaded TBT while respecting the maximum stress ( $\vartheta$ ) constraint on each of the three bars to adjust the cross-sectional areas, where  $B_1 (=x_1)$  and  $B_2 (=x_2)$ . Furthermore, the mathematical formulation of TBT problem is given as follows:

$$\text{Minimize: } f(x) = (2\sqrt{2}x_1 + x_2) \times L,$$

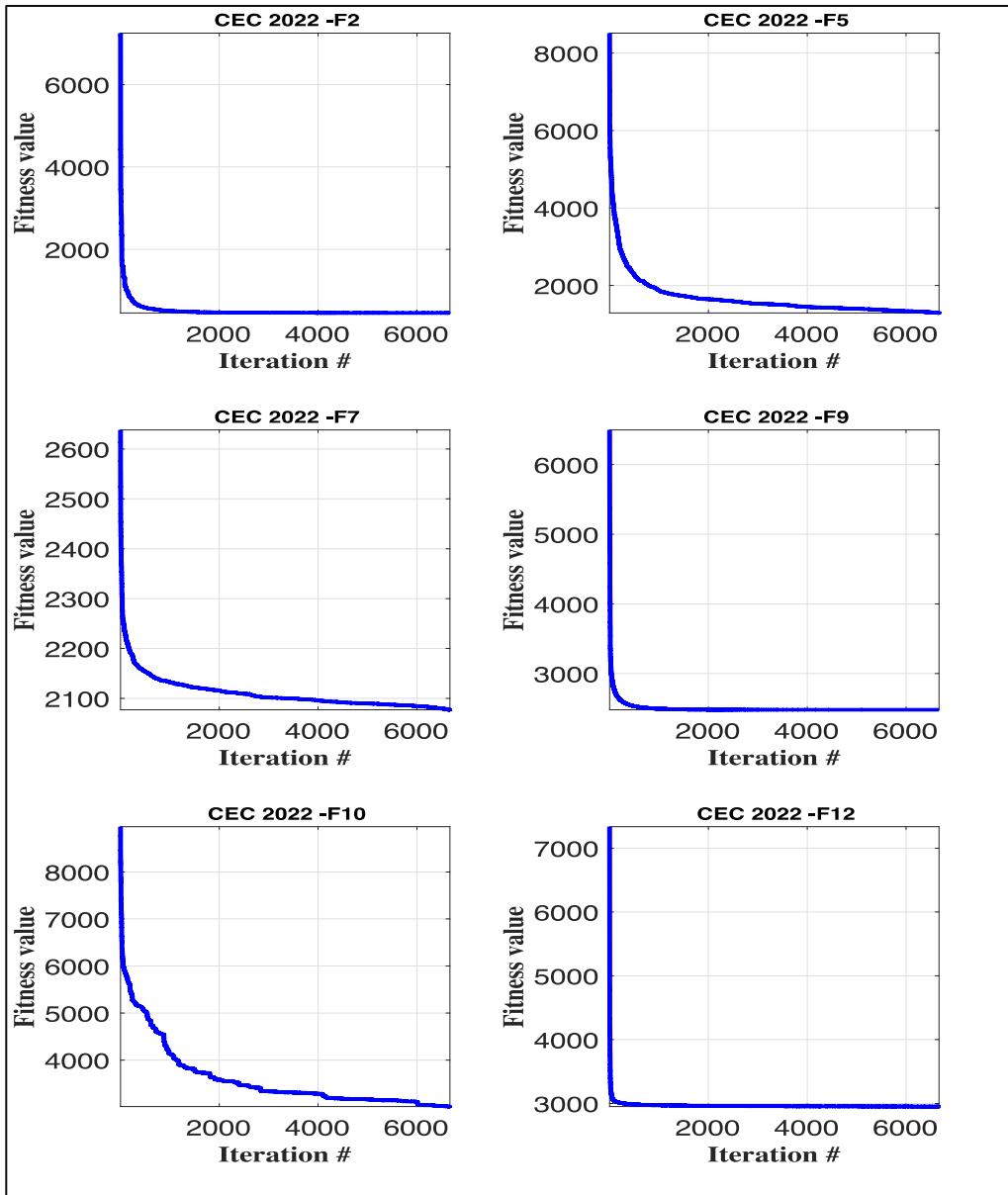
Subject to:

$$g_1(x) = \frac{(\sqrt{2}x_1 + x_2)}{\sqrt{2}x_1^2 + 2x_1x_2} P - \vartheta \leq 0,$$

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

$$g_3(x) = \frac{1}{\sqrt{2}x_2 + x_1} P - \vartheta \leq 0,$$

$$L = 100 \text{ cm},$$



**Fig. 30.** The convergence curve of YDSE over some CEC 2022 test functions using dimension = 20.

$$P = 2 \text{ kN/cm}^3,$$

$$\vartheta = 2 \text{ kN/cm}^3,$$

Variable range:

$$0 \leq x_1,$$

$$x_2 \leq 1.$$

Table 19 presents the optimized values for the three decision variables as well as the constraints and the objective function values for the TBT design problem. Table 20 performs a comparison between YDSE optimizer and other

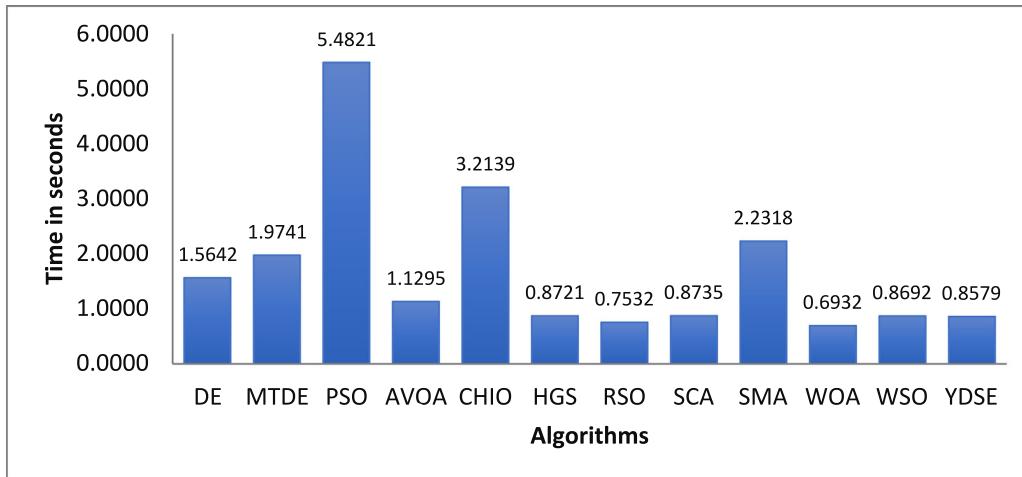


Fig. 31. The average CPU time over twelve test functions of the CEC 2022 benchmark.

Table 17

A comparison between YDSE optimizer and the winner of the CEC 2014 competition.

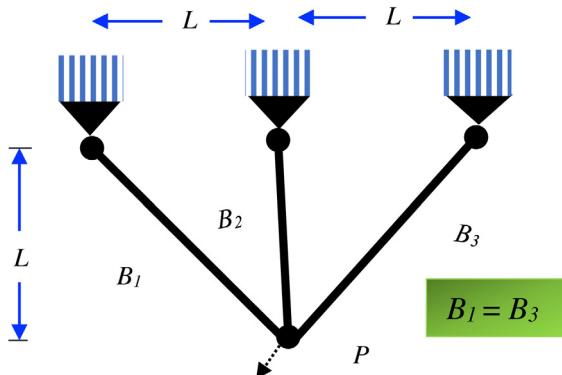
Fn.	Measure	L-SHADE	YDSEO	Fn.	Measure	L-SHADE	YDSEO
F1	Avg	1.0000E+02	<b>1.0000E+02</b>	F16	Avg	<b>1.6014E+03</b>	1.6024E+03
	SD	1.3712E-14	<b>0.0000E+00</b>		SD	<b>3.7142E-01</b>	2.6186E-01
F2	Avg	2.0000E+02	<b>2.0000E+02</b>	F17	Avg	1.7685E+03	<b>1.7025E+03</b>
	SD	7.4639E-15	<b>0.0000E+00</b>		SD	9.4043E+01	<b>2.0172E+00</b>
F3	Avg	3.0000E+02	<b>3.0000E+02</b>	F18	Avg	1.8027E+03	<b>1.8003E+03</b>
	SD	2.1111E-14	<b>0.0000E+00</b>		SD	3.1307E+00	<b>3.2818E-01</b>
F4	Avg	<b>4.0000E+02</b>	<b>4.0000E+02</b>	F19	Avg	1.9004E+03	<b>1.9008E+03</b>
	SD	<b>0.0000E+00</b>	<b>0.0000E+00</b>		SD	4.4582E-01	<b>3.8195E-01</b>
F5	Avg	<b>5.1877E+02</b>	5.2001E+02	F20	Avg	2.0006E+03	<b>2.0004E+03</b>
	SD	4.6540E+00	9.5922E-03		SD	9.1836E-01	<b>4.7270E-01</b>
F6	Avg	<b>6.0016E+02</b>	6.0082E+02	F21	Avg	2.1248E+03	<b>2.1004E+03</b>
	SD	<b>2.9156E-01</b>	8.4153E-01		SD	4.1886E+01	<b>2.5490E-01</b>
F7	Avg	7.0002E+02	<b>7.0002E+02</b>	F22	Avg	2.2281E+03	<b>2.2176E+03</b>
	SD	1.2680E-02	<b>1.2229E-02</b>		SD	3.1906E+01	<b>5.2163E+00</b>
F8	Avg	<b>8.0000E+02</b>	8.0183E+02	F23	Avg	2.6626E+03	<b>2.5911E+03</b>
	SD	<b>0.0000E+00</b>	7.7171E-01		SD	5.2054E+01	<b>1.6540E+13</b>
F9	Avg	9.0560E+02	<b>9.0517E+02</b>	F24	Avg	2.5141E+03	<b>2.5112E+03</b>
	SD	2.0982E+00	<b>1.9271E+00</b>		SD	1.6802E+01	<b>3.2239E+00</b>
F10	Avg	<b>1.0001E+03</b>	1.0446E+03	F25	Avg	2.6917E+03	<b>2.6599E+03</b>
	SD	<b>6.2131E-02</b>	3.8090E+01		SD	2.0611E+01	2.4510E+01
F11	Avg	<b>1.3044E+03</b>	1.5072E+03	F26	Avg	<b>2.7001E+03</b>	2.7001E+03
	SD	<b>8.7154E+01</b>	1.6281E+02		SD	<b>3.1489E-02</b>	4.3788E-02
F12	Avg	<b>1.2001E+03</b>	1.2001E+03	F27	Avg	3.0165E+03	<b>2.9173E+03</b>
	SD	<b>2.9027E-02</b>	5.8684E-02		SD	7.0982E+01	<b>1.6824E+02</b>
F13	Avg	1.3001E+03	<b>1.3001E+03</b>	F28	Avg	3.1388E+03	<b>3.1373E+03</b>
	SD	3.0977E-02	<b>2.3859E-02</b>		SD	3.3924E+00	<b>1.4740E+00</b>
F14	Avg	1.4002E+03	<b>1.4001E+03</b>	F29	Avg	3.1185E+03	<b>3.1124E+03</b>
	SD	6.2383E-02	<b>3.1146E-02</b>		SD	2.1325E+01	<b>1.1058E-01</b>
F15	Avg	<b>1.5004E+03</b>	1.5006E+03	F30	Avg	3.3334E+03	<b>3.2573E+03</b>
	SD	<b>9.2201E-02</b>	1.6916E-01		SD	7.4694E+01	<b>6.2651E+00</b>

MHs, such as AVOA, CHIO, GTO [109], RUN [56], PSO, SMA, HGS, WOA, WSO, RSO, and SCA. We can see that YDSE gets the same optimal value of the bar volume to be 2.6390E+02 for the best, worst, and mean cases.

**Table 18**

A comparison between YDSE optimizer and other winners of the CEC 2017 competition.

Fn.	Measure	LSHADE-cnEpSin	LSHADE-SPACMA	YDSE	Fn.	Measure	LSHADE-cnEpSin	LSHADE-SPACMA	YDSE
F1	Avg	<b>1.0000E+02</b>	<b>1.0000E+02</b>	<b>1.0000E+02</b>	F17	Avg	1.7003E+03	<b>1.7002E+03</b>	1.7218E+03
	SD	<b>0.0000E+00</b>	<b>0.0000E+00</b>	<b>0.0000E+00</b>		SD	6.8357E-01	<b>1.9278E-01</b>	6.5642E+00
F3	Avg	<b>3.0000E+02</b>	<b>3.0000E+02</b>	<b>3.0000E+02</b>	F18	Avg	1.8004E+03	1.8005E+03	<b>1.8003E+03</b>
	SD	<b>0.0000E+00</b>	<b>0.0000E+00</b>	<b>0.0000E+00</b>		SD	2.8722E-01	7.2950E+00	<b>3.0637E-01</b>
F4	Avg	<b>4.0000E+02</b>	<b>4.0000E+02</b>	<b>4.0000E+02</b>	F19	Avg	<b>1.9000E+03</b>	1.9003E+03	1.9004E+03
	SD	<b>0.0000E+00</b>	<b>0.0000E+00</b>	<b>0.0000E+00</b>		SD	<b>3.5833E-02</b>	5.1918E-01	3.5469E-01
F5	Avg	<b>5.0162E+02</b>	5.0178E+02	5.0621E+02	F20	Avg	2.0003E+03	<b>2.0003E+03</b>	2.0115E+03
	SD	<b>6.6581E-01</b>	6.9856E-01	2.4329E+00		SD	2.7189E-01	<b>1.3113E-01</b>	8.9695E+00
F6	Avg	<b>6.0000E+02</b>	<b>6.0000E+02</b>	6.0000E+02	F21	Avg	2.2618E+03	2.2029E+03	<b>2.2000E+03</b>
	SD	<b>0.0000E+00</b>	<b>0.0000E+00</b>	2.6295E-03		SD	5.1389E+01	1.4546E+01	<b>4.0498E-13</b>
F7	Avg	7.1197E+02	<b>7.1098E+02</b>	7.1570E+02	F22	Avg	2.3000E+03	2.3000E+03	<b>2.2663E+03</b>
	SD	6.5144E-01	<b>3.7398E-01</b>	2.1072E+00		SD	0.0000E+00	9.7949E-02	4.3502E+01
F8	Avg	8.0192E+02	<b>8.0107E+02</b>	805.5729782	F23	Avg	<b>2.6021E+03</b>	2.6022E+03	2.6083E+03
	SD	8.2329E-01	<b>8.1679E-01</b>	2.388506846		SD	<b>1.6524E+00</b>	1.5790E+00	3.0164E+00
F9	Avg	<b>9.0000E+02</b>	<b>9.0000E+02</b>	<b>9.0000E+02</b>	F24	Avg	2.6833E+03	2.6888E+03	<b>2.5788E+03</b>
	SD	<b>0.0000E+00</b>	<b>0.0000E+00</b>	<b>0.0000E+00</b>		SD	9.3240E+01	8.4986E+01	1.1332E+02
F10	Avg	1.0427E+03	<b>1.0177E+03</b>	1.3402E+03	F25	Avg	2.9160E+03	2.9211E+03	<b>2.8861E+03</b>
	SD	5.5259E+01	<b>3.1703E+01</b>	1.2708E+02		SD	2.2619E+01	2.2902E+01	5.7607E+01
F11	Avg	<b>1.1000E+03</b>	<b>1.1000E+03</b>	1.1001E+03	F26	Avg	2.9000E+03	2.9000E+03	<b>2.7833E+03</b>
	SD	<b>0.0000E+00</b>	<b>0.0000E+00</b>	2.5243E-01		SD	0.0000E+00	0.0000E+00	1.2058E+02
F12	Avg	1.2892E+03	1.338E+03	<b>1.2044E+03</b>	F27	Avg	3.0894E+03	3.0896E+03	<b>3.0890E+03</b>
	SD	5.6579E+01	7.4253E+01	<b>2.1636E+01</b>		SD	1.9616E+00	7.9605E-01	<b>5.6423E-01</b>
F13	Avg	1.3038E+03	1.3039E+03	<b>1.3018E+03</b>	F28	Avg	3.1362E+03	3.1116E+03	<b>3.0932E+03</b>
	SD	2.5449E+00	2.4959E+00	<b>2.0269E+00</b>		SD	9.4131E+01	5.8444E+01	<b>5.8116E+01</b>
F14	Avg	<b>1.4000E+03</b>	<b>1.4000E+03</b>	1.4011E+03	F29	Avg	<b>3.1284E+03</b>	3.1314E+03	3.1357E+03
	SD	<b>1.8165E-01</b>	<b>4.3775E-01</b>	1.1338E+00		SD	<b>1.9022E+00</b>	2.9782E+00	1.8692E+01
F15	Avg	1.5003E+03	1.5003E+03	<b>1.5002E+03</b>	F30	Avg	5.7346E+03	5.1499E+04	<b>3.3970E+03</b>
	SD	2.1072E-01	2.5233E-01	<b>1.7519E-01</b>		SD	1.2699E+04	1.9418E+05	<b>9.2097E+00</b>
F16	Avg	1.6007E+03	<b>1.6007E+03</b>	1.6013E+03					
	SD	3.3202E-01	<b>2.6356E-01</b>	5.8299E-01					

**Fig. 32.** Three-bar truss design problem.

The results in the table demonstrate the superiority of YDSE over the other algorithms with a minimum standard deviation value of 0.0000E+00. YDSE takes an average time of 1.94 s to tackle the problem, which is smaller compared to CHIO, GTO, RUN, PSO, SMA, HGS, WSO, RSO, and SCA. Even though the WOA algorithm can solve the issue in 1.37 s, the optimal volume values reached are lower than YDSE's values, as can be concluded from the SD value of 3.7050E-02.

**Table 19**

The best results obtained by YDSE on the three-bar truss design problem.

	Symbols	Value
Variable	$x_1$	7.8868E-01
	$x_2$	4.0825E-01
Constraints	$g_1(x)$	-2.2204E-15
	$g_2(x)$	-1.4641
	$g_3(x)$	-0.5359
Objective function	$f(x)$	2.6390E+02

**Table 20**

Results of the algorithms for three-bar truss design problem.

Algorithm	Worst	Mean	Best	SD	CPU time
AVOA	2.6390E+02	2.6390E+02	2.6390E+02	3.3925E-05	1.49
CHIO	2.6396E+02	2.6391E+02	2.6390E+02	1.4595E-02	2.36
GTO	2.6390E+02	2.6390E+02	2.6390E+02	2.6312E-09	3.34
RUN	2.6390E+02	2.6390E+02	2.6390E+02	1.4752E-05	4.74
PSO	2.6390E+02	2.6390E+02	2.6390E+02	1.3762E-07	3.25
SMA	2.7073E+02	2.6693E+02	2.6422E+02	2.1817E+00	2.10
HGS	2.7004E+02	2.6524E+02	2.6391E+02	1.4835E+00	1.78
WOA	2.6403E+02	2.6392E+02	2.6390E+02	3.7050E-02	1.37
WSO	2.6390E+02	2.6390E+02	2.6390E+02	1.0556E-14	1.83
RSO	2.7421E+02	2.6816E+02	2.6417E+02	2.7015E+00	1.72
SCA	2.6392E+02	2.6391E+02	2.6390E+02	6.4056E-03	1.50
<b>YDSE</b>	<b>2.6390E+02</b>	<b>2.6390E+02</b>	<b>2.6390E+02</b>	<b>0.0000E+00</b>	<b>1.49</b>

The **Bold** font indicate the best results.

### 6.8.2. Tension/compression spring design

The structure of the tension/compression spring is illustrated in Fig. 33. The fitness function of this problem is to reduce the weight of the spring subject to four constraints on shear stress minimum deflection, surge frequency, and limits on the outside diameter. The problem seeks to optimize the values of three parameters (mean coil diameter  $D (=x_1)$ , wire diameter  $d (=x_2)$ , number of active coils  $L (=x_3)$ ). The formulation of the problem is as follows:

*Minimize :*

$$f(x) = (x_3 + 2).x_2.x_1^2,$$

*Subject to :*

$$\begin{aligned} g_1(x) &= 1 - \frac{x_2 \cdot x_3}{71785x_1^4} \leq 0, \\ g_2(x) &= 1 - \frac{4x_2^2 - x_1 \cdot x_2}{12566(x_2 \cdot x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0, \\ g_3(x) &= 1 - \frac{140.45x_1}{x_2^2 \cdot x_3} \leq 0, \\ g_4(x) &= \frac{x_1 + x_2}{1.5} - 1 \leq 0, \end{aligned} \quad (29)$$

*Variable range :*

$$0.05 \leq x_1 \leq 2,$$

$$0.25 \leq x_2 \leq 1.3,$$

$$2 \leq x_3 \leq 15.$$

Table 21 presents the optimized values for the three decision variables as well as the constraints and the objective function values for the tension/compression spring design problem. Moreover, the performance of the proposed

**Table 21**

The best results obtained by YDSE on the tension/compression spring design problem.

	Symbols	Value
Variable	$x_1$	5.1689E-02
	$x_2$	3.5673E-01
	$x_3$	1.1288E+01
Constraints	$g_1(x)$	-6.8306E-11
	$g_2(x)$	-1.5262E-11
	$g_3(x)$	-4.0538
	$g_4(x)$	-0.7277
Objective function	$f(x)$	1.2665E-02

**Table 22**

Results of the algorithms for tension/compression spring design problem.

Algorithm	Worst	Mean	Best	SD	CPU time
AVOA	1.3812E-02	1.3001E-02	1.2667E-02	3.8188E-04	1.58
CHIO	1.3454E-02	1.2881E-02	1.2670E-02	2.0080E-04	2.35
GTO	1.2719E-02	1.2689E-02	1.2667E-02	1.8850E-05	3.23
RUN	1.7773E-02	1.3019E-02	1.2665E-02	1.0413E-03	7.62
PSO	1.3017E-02	1.2753E-02	1.2665E-02	9.1415E-05	3.45
SMA	1.3259E-02	1.2794E-02	1.2676E-02	1.7786E-04	2.22
HGS	2.8528E+00	1.7813E+00	1.7249E+00	2.0988E-01	1.33
WOA	1.5585E-02	1.3438E-02	1.2666E-02	9.1406E-04	1.55
WSO	1.4434E+00	6.0375E-02	1.2665E-02	2.6122E-01	1.56
RSO	2.7935E+00	1.5535E+00	1.2842E-02	1.0687E+00	1.83
SCA	1.3067E-02	1.2901E-02	1.2738E-02	8.7128E-05	1.88
<b>YDSE</b>	<b>1.2665E-02</b>	<b>1.2665E-02</b>	<b>1.2665E-02</b>	<b>1.5518E-09</b>	<b>1.56</b>

YDSE is tested against ten MH algorithms in terms of worst, mean, best, SD, and CPU time. The results of this experiment are listed in [Table 22](#). The statistical results of the algorithms prove that YDSE has the minimum weight of the spring for the best, mean, and worst cases with smaller SD compared to those obtained by the other algorithms. Although, WOA and HGS have the minimum CPU times of 1.33 and 1.55, the SD values are higher than YDSE with 2.0988E-01 and 9.1406E-04, respectively.

### 6.8.3. Cantilever beam design problem

Generally, this EOP consists of five hollow square pieces that own the same thickness and boosted from the first piece, whereas the other pieces are free, as shown in [Fig. 34](#). The objective is to minimize the weight of the beam, whereas the beam's fifth piece is affected by a vertical power. The lengths of the five pieces represent the five decision variables ( $x_1, x_2, x_3, x_4, x_5$ ) having only one constraint. The equation can be formulated as in Eq. (30).

*Minimize:*

$$f(x) = 0.0624 \times (x_1 + x_2 + x_3 + x_4 + x_5) \times L,$$

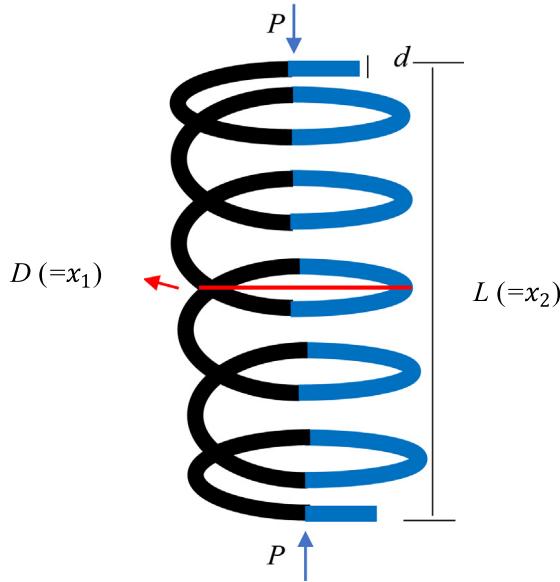
*Subject to:*

$$g(x) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0, \quad (30)$$

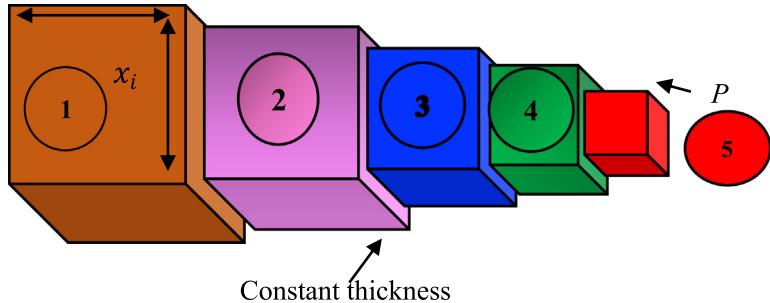
*Variable range:*

$$0.01 \leq x_i \leq 100, \quad i = 1, 2, \dots, 5.$$

In [Table 23](#), the optimum results of YDSE are presented for five decision variables, one constraint and the objective function. [Table 24](#) displays the results of eleven algorithms for solving the cantilever beam design problem. By observing the results, YDSE can beat its rivals in minimizing the weight of the beam with a best value equal to



**Fig. 33.** Tension/compression spring design.



**Fig. 34.** Cantilever beam design problem.

**Table 23**

The best results obtained by YDSE on the Cantilever beam design problem.

	Symbols	Value
Variable	$x_1$	6.0160E+00
	$x_2$	5.3092E+00
	$x_3$	4.4943E+00
	$x_4$	3.5015E+00
	$x_5$	2.1527E+00
Constraints	$g_1(x)$	0
Objective function	$f(x)$	1.3400E+00

1.3400E+00 accompanied with a minimum SD value of 4.4024E–16. The second ranked algorithm is WSO that achieves good results with a SD value of 4.5168E–16. Even though algorithms like AVOA, RUN, PSO, HGS, and SMA produced satisfactory results, they are cumbersome in terms of the CPU time.

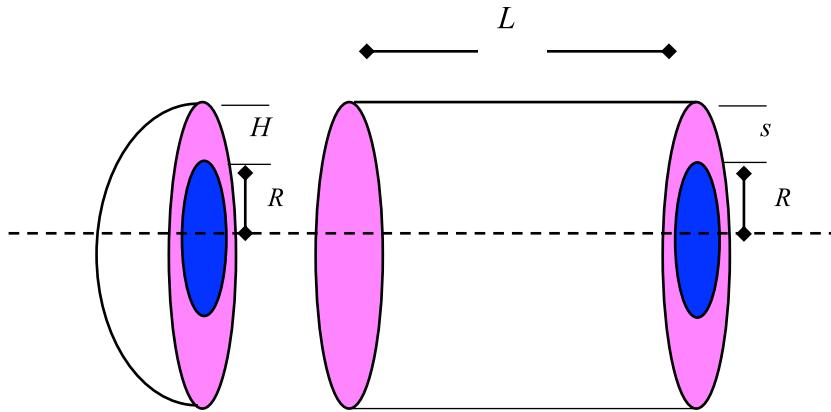
#### 6.8.4. Pressure vessel design problem

The design of pressure vessel is another EOP that aims to lessen the total cost of the cylindrical pressure vessel, including cost of material, welding, and forming. This problem contains 4 decision variables: Shell thickness  $S(=x_1)$ ,

**Table 24**

Results of the algorithms for cantilever beam design problem.

Algorithm	Worst	Mean	Best	SD	CPU time
AVOA	1.3400E+00	1.3400E+00	1.3400E+00	1.2995E-05	1.19
CHIO	1.5840E+00	1.4102E+00	1.3491E+00	6.0234E-02	3.08
GTO	1.3400E+00	1.3400E+00	1.3400E+00	1.1479E-05	2.44
RUN	1.3400E+00	1.3400E+00	1.3400E+00	4.6939E-07	4.17
PSO	1.3400E+00	1.3400E+00	1.3400E+00	7.5083E-07	2.78
SMA	1.3400E+00	1.3400E+00	1.3400E+00	5.7483E-05	1.60
HGS	1.3401E+00	1.3400E+00	1.3400E+00	3.7928E-05	1.10
WOA	1.5166E+00	1.3821E+00	1.3486E+00	3.2068E-02	1.00
WSO	1.3400E+00	1.3400E+00	1.3400E+00	4.5168E-16	1.03
RSO	1.4726E+00	1.3899E+00	1.3449E+00	3.2725E-02	1.06
SCA	1.4042E+00	1.3705E+00	1.3467E+00	1.3884E-02	1.31
<b>YDSE</b>	<b>1.3400E+00</b>	<b>1.3400E+00</b>	<b>1.3400E+00</b>	<b>4.4024E-16</b>	1.03

**Fig. 35.** The structure of pressure vessel design.

head thickness  $H(=x_2)$ , inner radius  $R(=x_3)$ , range of cross-section minus head  $M(=x_4)$ . [Fig. 35](#) depicts the structure of pressure vessel design problem. Moreover,  $S$  and  $H$  are integer values multiple of 0.0625, while the remaining variables are continuous. The mathematical equations is expressed as below:

*Minimize:*

$$f(x) = 0.6224 \times x_1 \cdot x_3 \cdot x_4 + 1.7781 \times x_2 \cdot x_3^2 + 3.1661 \times x_1^2 \cdot x_4 + 19.84x_1^2 x_3,$$

*Subject to:*

$$g_1(x) = -x_1 + 0.0193x_3 \leq 0,$$

$$g_2(x) = -x_2 + 0.00954x_3 \leq 0,$$

$$g_3(x) = -\pi x_3^2 \cdot x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0, \quad (31)$$

$$g_4(x) = -x_4 - 240 \leq 0,$$

*Variable range:*

$$x_1, x_2 \in \{1 \times 0.0625, 2 \times 0.0625, \dots, 1600 \times 0.0625\},$$

$$10 \leq x_3,$$

$$x_4 \leq 200.$$

[Table 25](#) provides the optimal results obtained by our proposed algorithm for four variables and four constraints. [Table 26](#) constructs a comparison among the algorithms. By inspecting the results, we can see that the performance

**Table 25**

The best results obtained by YDSE on the pressure vessel design problem.

	Symbols	Value
Variable	$x_1$	1.3233E+01
	$x_2$	7.0746E+00
	$x_3$	4.2098E+01
	$x_4$	1.7664E+02
Constraints	$g_1(x)$	-12.4200
	$g_2(x)$	-6.6730
	$g_3(x)$	0
	$g_4(x)$	-63.3634
Objective function	$f(x)$	6.0597E+03

**Table 26**

Results of the algorithms for pressure vessel design problem.

Algorithm	Worst	Mean	Best	SD	CPU time
AVOA	7.3096E+03	6.6074E+03	6.0597E+03	4.6996E+02	1.63
CHIO	6.5798E+03	6.2724E+03	6.0741E+03	1.4672E+02	2.40
GTO	6.0905E+03	6.0684E+03	6.0597E+03	8.7913E+00	3.38
RUN	7.3097E+03	6.2740E+03	6.0597E+03	4.5368E+02	7.20
PSO	7.6358E+03	6.1842E+03	6.0597E+03	3.2032E+02	3.58
SMA	7.0475E+03	6.2578E+03	6.0597E+03	3.0679E+02	2.27
HGS	7.3096E+03	6.5970E+03	6.0598E+03	5.3674E+02	1.69
WOA	7.8046E+03	6.8851E+03	6.0836E+03	6.1362E+02	1.40
WSO	7.5803E+03	6.5219E+03	6.0597E+03	5.3950E+02	1.62
RSO	3.3146E+04	1.6308E+04	7.5685E+03	6.1215E+03	1.75
SCA	7.6977E+03	6.4334E+03	6.0816E+03	4.1764E+02	1.60
<b>YDSE</b>	<b>6.0597E+03</b>	<b>6.0597E+03</b>	<b>6.0597E+03</b>	<b>9.2504E-13</b>	<b>1.46</b>

of most of competitors is degraded. The results show that YDSE can attain the minimum best, mean, and worst values with cost value 6.0597E+03 and a SD with a value of 9.2504E-13. The quality of the outcomes of other algorithms is not good accompanied with high SD values.

#### 6.8.5. Speed reducer design problem

Generally, the speed reducer design [110] is regarded a challenging EOP as we need to keep the weight of the speed reducer as low as possible while meeting a total of eleven constraints. This problem typically has 7 decision variables to be optimized, including face width  $Fw(=x_1)$ , teeth module  $Tm(=x_2)$ , number of teeth on pinion  $Tn(=x_3)$ , the first shaft length between bearings  $Ls_1(=x_4)$ , the second shaft length between bearings  $Ls_2(=x_5)$ , the first shaft diameter  $Sd_1(=x_6)$  and the second shaft diameter  $Sd_2(=x_7)$ . Fig. 36 presents a description for the speed design problem. This problem can be formulated as:

*Minimize:*

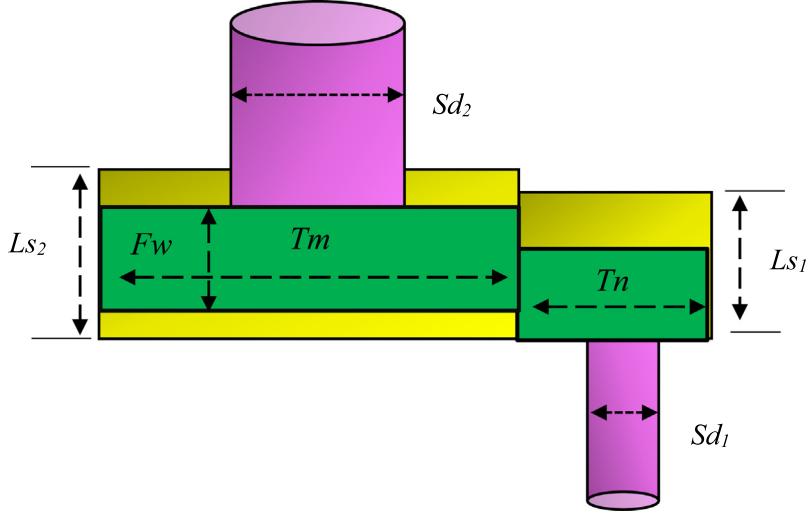
$$f(x) = 0.7854 \times x_1 \cdot x_2^2 (3.3333 \times x_3^2 + 14.933 \times 4x_3 - 43.0934) \\ - 1.508x_1 \times (x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3) + 0.7854(x_4 \cdot x_6^2 + x_5 \cdot x_7^2),$$

*subject to:*

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

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

$$g_3(x) = \frac{1.93 \times x_4^3}{x_2 \cdot x_6^4 \cdot x_3} - 1 \leq 0,$$



**Fig. 36.** The structure of speed reducer design problem.

$$\begin{aligned}
 g_4(x) &= \frac{1.93 \times x_5^3}{x_2 \cdot x_7^4 \cdot x_3} - 1 \leq 0, \\
 g_5(x) &= \frac{\sqrt{(745x_4/x_2x_3)^2 + 16.9 \times 10^6}}{110x_6^3} - 1 \leq 0, \\
 g_6(x) &= \frac{\sqrt{(745x_5/x_2x_3)^2 + 157.5 \times 10^6}}{85x_7^3} - 1 \leq 0, \\
 g_7(x) &= \frac{x_2 \cdot x_3}{40} - 1 \leq 0, \\
 g_8(x) &= \frac{5x_2}{x_1} - 1 \leq 0, \\
 g_9(x) &= \frac{x_1}{12x_2} - 1 \leq 0, \\
 g_{10}(x) &= \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0, \\
 g_{11}(x) &= \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0,
 \end{aligned} \tag{32}$$

Variable range:

$$2.6 \leq x_1 \leq 3.6,$$

$$0.7 \leq x_2 \leq 0.8,$$

$$x_3 \in \{17, 18, \dots, 28\},$$

$$7.3 \leq x_4,$$

$$x_5 \leq 8.3,$$

$$2.9 \leq x_6 \leq 3.9,$$

$$5 \leq x_7 \leq 5.5.$$

**Table 27**

The best results obtained by YDSE on the speed reducer design problem.

	Symbols	Value
Variable	$x_1$	3.5000E+00
	$x_2$	7.0000E-01
	$x_3$	1.7000E+01
	$x_4$	7.3000E+00
	$x_5$	7.7153E+00
	$x_6$	3.3505E+00
	$x_7$	5.2867E+00
Constraints	$g_1(x)$	-2.1550
	$g_2(x)$	-98.1350
	$g_3(x)$	-1.9251
	$g_4(x)$	-18.3099
	$g_5(x)$	-3.8654E-12
	$g_6(x)$	-1.3642E-12
	$g_7(x)$	-28.1000
	$g_8(x)$	3.5527E-15
	$g_9(x)$	-7.0000
	$g_{10}(x)$	-0.3742
	$g_{11}(x)$	3.1086E-15
Objective function	$f(x)$	2.9944E+03

**Table 28**

Results of the algorithms for designing problem of speed reducer.

Algorithm	Worst	Mean	Best	SD	CPU time
AVOA	2.9947E+03	2.9944E+03	2.9944E+03	6.4318E-02	1.81
CHIO	2.9952E+03	2.9946E+03	2.9944E+03	1.9956E-01	2.54
GTO	3.0153E+03	2.9963E+03	2.9944E+03	5.0312E+00	3.88
RUN	3.0104E+03	3.0012E+03	2.9950E+03	4.3155E+00	7.80
PSO	3.0060E+03	2.9958E+03	2.9944E+03	3.6349E+00	3.41
SMA	2.9944E+03	2.9944E+03	2.9944E+03	4.5876E-03	2.34
HGS	<b>2.9944E+03</b>	<b>2.9944E+03</b>	<b>2.9944E+03</b>	<b>1.0513E-12</b>	1.83
WOA	3.0944E+03	3.0285E+03	2.9989E+03	2.5487E+01	1.72
WSO	3.3379E+03	3.0161E+03	2.9944E+03	6.3066E+01	1.87
RSO	4.0855E+03	3.5472E+03	3.2026E+03	2.3075E+02	1.78
SCA	3.0829E+03	3.0511E+03	3.0101E+03	1.8198E+01	1.67
YDSE	2.9944E+03	2.9944E+03	2.9944E+03	7.7781E-10	1.81

**Table 27** records the best results obtained by YDSE optimizer. **Table 28** presents the results of ten algorithms for tackling the speed reducer design problem. We can see that HGS has the minimum weight value of the problem with a value of 2994.245 and SD = 1.0513E-12. Also, YDSE is ranked the second algorithm with minimum weight value = 2.9944E+03 and SD = 7.7781E-10, while RSO achieves the worst results.

#### 6.8.6. Gear train design problem

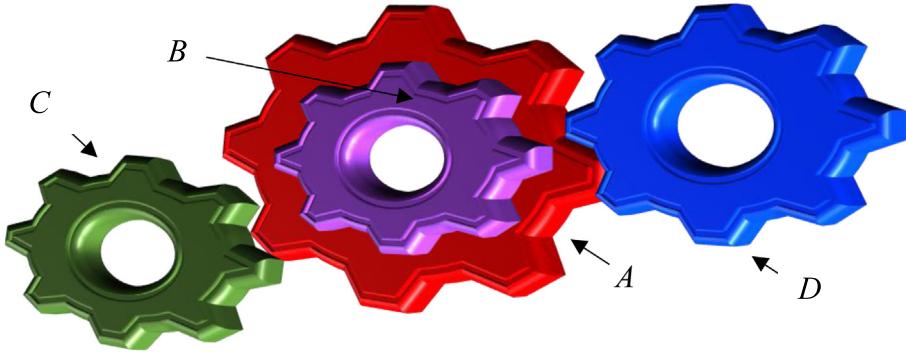
This subsection verifies the ability of YDSE for tackling the design problem of gear train, which is regarded as unconstrained discrete EOP. [Fig. 37](#) displays a visual illustration for this problem. The goal is to find the number of teeth for each one of the four wheels: A(=x<sub>1</sub>), B(=x<sub>2</sub>), C(=x<sub>3</sub>), and D(=x<sub>4</sub>) of the gear to minimize the gear ratio.

$$\text{Minimize: } f(x) = \left( \frac{1}{6.931} - \frac{x_3 x_2}{x_1 x_4} \right)^2, \quad (33)$$

Variable range:

$$x_1, x_2, x_3, x_4, x_5 \in \{12, 13, \dots, 60\}.$$

**Table 29** introduces the results obtained of YDSE. For further investigating the performance of YDSE, it is employed for addressing the gear train design problem and tested against other algorithms. Furthermore, the best



**Fig. 37.** The structure of gear train design problem.

**Table 29**

The best results obtained by YDSE on the structure of gear train design problem.

	Symbols	Value
Variable	$x_1$	4.2927E+01
	$x_2$	1.8914E+01
	$x_3$	1.5530E+01
	$x_4$	4.9234E+01
Objective function	$f(x)$	<b>2.7009E-12</b>

**Table 30**

The results of the algorithms for the design of gear train.

Algorithm	Worst	Mean	Best	SD	CPU time
AVOA	2.3576E-09	5.6795E-10	2.7009E-12	6.2703E-10	1.09
CHIO	2.6140E-08	4.7934E-09	2.3078E-11	6.4985E-09	1.85
GTO	8.8876E-10	7.2467E-11	2.7009E-12	2.2269E-10	2.50
RUN	1.3616E-09	2.0438E-10	2.7009E-12	3.9071E-10	3.50
PSO	8.7008E-09	1.2654E-09	2.7009E-12	2.1467E-09	2.63
SMA	2.7265E-08	1.9005E-09	2.3078E-11	4.8380E-09	1.54
HGS	2.7265E-08	3.7641E-09	8.8876E-10	6.5989E-09	1.60
WOA	2.3576E-09	8.3214E-10	2.7009E-12	8.3196E-10	0.93
WSO	2.5670E-07	2.4486E-08	9.9399E-11	6.3543E-08	0.94
RSO	2.8026E-02	3.5106E-03	1.8274E-08	6.9416E-03	0.93
SCA	2.0575E-09	6.5388E-10	2.7009E-12	5.6923E-10	0.93
<b>YDSE</b>	<b>2.3078E-11</b>	<b>1.3569E-11</b>	<b>2.7009E-12</b>	<b>6.26E-11</b>	<b>0.95</b>

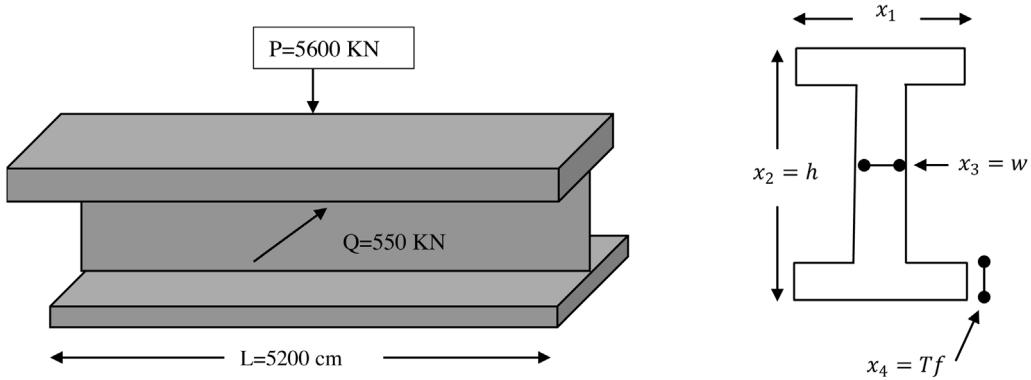
results of eleven algorithms are provided in **Table 30**. It is clear that YDSE outperforms overall algorithms for the best, mean, worst values with values 2.7009E-12, 1.3569E-11 and 2.3078E-11, respectively. Additionally, YDSE has the smallest SD value of 6.26E-11 and acceptable CPU time of 0.95 s.

#### 6.8.7. I-beam vertical deflection

Another EOP is the design of I-beam that minimizes the vertical deflection of the I-beam subject a design load ( $P$ ) and a horizontal load ( $Q$ ) acting at midspan, as seen in **Fig. 38**. The vertical deflection of the I-beam can be computed as  $(PL^3)/(48EI)$ .  $L$  is the length of the I-beam and  $E$  refers to the modulus of elasticity. Four decision variables are optimized and represent the width of flange  $x_1 (=w)$ , the height of the section  $x_2 (=h)$ , the thickness of the web  $x_3 (=Tw)$ , and the thickness of the flange  $x_4 (=Tf)$ . We can express the problem as:

*Minimize:*

$$f(x) = \frac{5000}{\frac{x_3 \times (x_2 - 2x_4)^3}{12} + \left(\frac{x_1 \cdot x_4^3}{6}\right) + 2b \cdot x_4(x_2 - \frac{x_4}{2})^2},$$

**Fig. 38.** The structure of I-beam vertical deflection.**Table 31**

The best results obtained by YDSE on the I-beam design problem.

	Symbols	Value
Variable	$x_1$	1.3233E+01
	$x_2$	7.0746E+00
	$x_3$	4.2098E+01
	$x_4$	1.7664E+02
Constraints	$g_1(x)$	-12.4200
	$g_2(x)$	-6.6730
	$g_3(x)$	0
	$g_4(x)$	-63.3634
Objective function	$f(x)$	8.8219E-18

*Subject to:*

$$g_1(x) = 2x_1 \cdot x_3 + x_3 \times (x_2 - 2x_4) \leq 300,$$

$$g_2(x) = \frac{18x_2 \times 10^4}{x_3(x_2 - 2x_4)^3 + 2x_1x_3(4x_4^2 + 3x_2(x_2 - 2x_4))} + \frac{15x_1 \times 10^3}{x_3^2(x_2 - 2x_4) + 2x_3 \cdot x_1^3} \leq 56,$$

*Variable range:*

$$10 \leq x_1 \leq 50,$$

$$10 \leq x_2 \leq 80,$$

$$0.9 \leq x_3 \leq 5,$$

$$0.9 \leq x_4 \leq 5.$$

(34)

**Table 31** contains the optimized values obtained by the proposed algorithm for four decision variables and two constraints. Furthermore, **Table 32** conducts a comparison among the algorithms using the I-beam design deflection problem. There are four algorithms: PSO, HGS, WSO and YDSE, which can reach the optimal values of the vertical deflection with small SD value = 8.8219E-18. Despite those algorithms achieve the same optimal values, YDSE surpasses its rivals as it has the smallest CPU time with a value of 1.29 s. However, PSO gets competitive results, it takes much time to solve the problem which is approximately twice of the proposed algorithm.

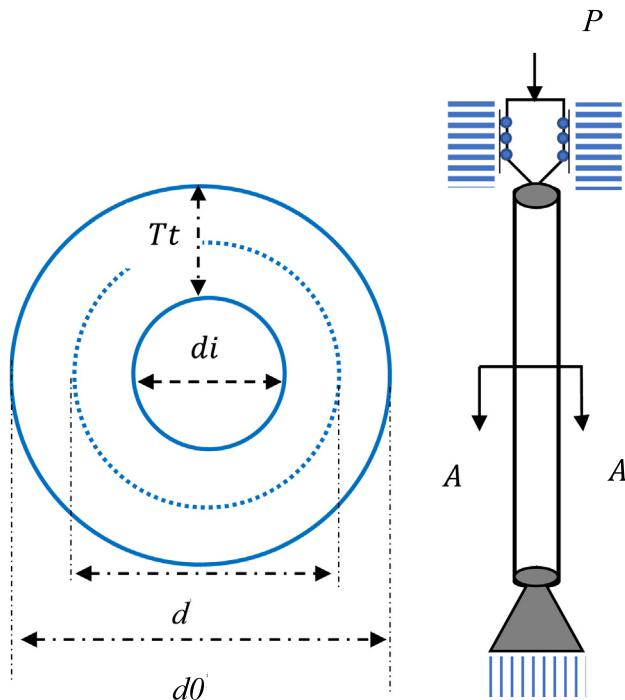
#### 6.8.8. Tubular column design

In this problem, the tubular section is design to carry the compressive load  $P$  at a minimum cost subject to six constraints, as depicted in **Fig. 39**. The column is affected by axial force  $P$ . The problem has two decision variables:

**Table 32**

The results of the algorithms for I-beam design deflection.

Algorithm	Worst	Mean	Best	SD	CPU time
AVOA	1.3074E-02	1.3074E-02	1.3074E-02	7.5541E-09	1.42
CHIO	1.4193E-02	1.3483E-02	1.3074E-02	2.9125E-04	2.21
GTO	1.3074E-02	1.3074E-02	1.3074E-02	8.1615E-14	2.82
RUN	1.3074E-02	1.3074E-02	1.3074E-02	6.2929E-10	4.68
<b>PSO</b>	<b>1.3074E-02</b>	<b>1.3074E-02</b>	<b>1.3074E-02</b>	<b>8.8219E-18</b>	3.08
SMA	1.3074E-02	1.3074E-02	1.3074E-02	1.2573E-08	1.97
<b>HGS</b>	<b>1.3074E-02</b>	<b>1.3074E-02</b>	<b>1.3074E-02</b>	<b>8.8219E-18</b>	1.54
WOA	1.3764E-02	1.3269E-02	1.3074E-02	1.9799E-04	1.44
<b>WSO</b>	<b>1.3074E-02</b>	<b>1.3074E-02</b>	<b>1.3074E-02</b>	<b>8.8219E-18</b>	1.33
RSO	7.4372E-01	2.7152E-01	1.7340E-02	1.9920E-01	1.48
SCA	1.3142E-02	1.3092E-02	1.3074E-02	1.3242E-05	1.63
<b>YDSE</b>	<b>1.3074E-02</b>	<b>1.3074E-02</b>	<b>1.3074E-02</b>	<b>8.8219E-18</b>	<b>1.29</b>

**Fig. 39.** The structure of tubular column and A–A cross-section design problem.

the central diameter of the column  $d(=x_1)$  and tube thickness  $Tt(=x_2)$ . The material of this column yields a stress of  $\sigma_y = 500 \text{ kgf/cm}^2$  and Elasticity modulus of  $E = 0.85 \times 10^6 \text{ kgf/cm}^2$ . The mathematical model is expressed as follows:

*Minimize:*

$$f(x) = 9.8x_1x_2 + 2x_1,$$

*Subject to:*

$$g_1(x) = \frac{P}{\pi x_1 x_2 \sigma_y} - 1 \leq 0,$$

$$g_2(x) = \frac{8PL^2}{\pi^3 E x_1 x_2 (x_1^2 + x_2^2)} - 1 \leq 0,$$

**Table 33**

The best results obtained by YDSE on the tubular column design problem.

	Symbols	Value
Variable	$x_1$	5.4522E+00
	$x_2$	2.9163E-01
Constraints	$g_1(x)$	-4.4409E-16
	$g_2(x)$	0
	$g_3(x)$	-0.6332
	$g_4(x)$	-0.6106
	$g_5(x)$	-0.6332
	$g_6(x)$	-0.3185
Objective function	$f(x)$	2.6486E+01

**Table 34**

Results of the algorithms for the tubular column design problem.

Algorithm	Worst	Mean	Best	SD	CPU time
AVOA	2.6486E+01	2.6486E+01	2.6486E+01	1.0344E-09	1.46
CHIO	2.7061E+01	2.6605E+01	2.6492E+01	1.3765E-01	2.24
GTO	2.6486E+01	2.6486E+01	2.6486E+01	1.3080E-08	2.88
RUN	2.6486E+01	2.6486E+01	2.6486E+01	1.3941E-06	4.70
<b>PSO</b>	<b>2.6486E+01</b>	<b>2.6486E+01</b>	<b>2.6486E+01</b>	<b>0.0000E+00</b>	3.43
SMA	2.6487E+01	2.6486E+01	2.6486E+01	6.4289E-05	1.97
HGS	2.6486E+01	2.6486E+01	2.6486E+01	2.1283E-12	1.48
WOA	2.6881E+01	2.6582E+01	2.6487E+01	8.9438E-02	1.70
<b>WSO</b>	<b>2.6486E+01</b>	<b>2.6486E+01</b>	<b>2.6486E+01</b>	<b>0.0000E+00</b>	1.87
RSO	3.8020E+01	2.9797E+01	2.6788E+01	3.1394E+00	1.56
SCA	2.6623E+01	2.6551E+01	2.6491E+01	3.3744E-02	1.47
<b>YDSE</b>	<b>2.6486E+01</b>	<b>2.6486E+01</b>	<b>2.6486E+01</b>	<b>0.0000E+00</b>	<b>1.43</b>

$$g_3(x) = \frac{2}{x_1} - 1 \leq 0,$$

$$g_4(x) = \frac{x_1}{14} - 1 \leq 0,$$

$$g_5(x) = \frac{0.2}{x_2} - 1 \leq 0,$$

$$g_6(x) = \frac{x_2}{8} - 1 \leq 0,$$

Variable range:

$$2 \leq x_1 \leq 14,$$

$$0.2 \leq x_2 \leq 0.8.$$

The best results obtained by the suggested YDSE for the two decision variables and six constraints can be found in [Table 33](#). For further testing the quality of our algorithm, YDSE is compared to ten MH algorithms as provided in [Table 34](#). According to the table, PSO, WSO, and YDSE all achieve the top rank with a minimum cost of 2.6486E+01 and SD of 0.000E+00. For the tubular column design, YDSE takes only 1.43 s on the CPU, which is faster than PSO and WSO. Overall, RSO shows the worst performance among the eleven algorithms with SD value of 3.1394E+00.

#### 6.8.9. Piston lever design problem

With this issue, we are trying to minimize the oil volume, while raising the piston lever from 0° to 45°, in order to find the four piston components,  $H (= x_1)$ ,  $B (= x_2)$ ,  $D (= x_3)$  and  $X (= x_4)$ , as can be seen in [Fig. 40](#). The

**Table 35**

The best results obtained by YDSE on the piston lever design problem.

	Symbols	Value
Variable	$x_1$	5.0000E-02
	$x_2$	2.0415E+00
	$x_3$	4.0830E+00
	$x_4$	1.2000E+02
Constraints	$g_1(x)$	-2.3283E-10
	$g_2(x)$	-6.0000E+05
	$g_3(x)$	-117.1875
	$g_4(x)$	-5.7732E-15
Objective function	$f(x)$	6.9128E-13

mathematical equations can be expressed as below:

*Minimize:*

$$f(x) = \frac{1}{4} \times \pi \times x_3^2 \times (L_2 - L_1),$$

*Subject to:*

$$g_1(x) = Q \times L \times \cos \theta - R \times F \leq 0,$$

$$g_2(x) = Q \times (L - x_4) - M_{\max} \leq 0,$$

$$g_3(x) = 1.2 \times (L_2 - L_1) - L_1 \leq 0,$$

$$g_4(x) = \frac{x_3}{2} - x_2 \leq 0,$$

$$R = \frac{|-x_4 \times (x_4 \times \sin \theta + x_1) + x_1 \times (x_2 - x_4 \times \cos \theta)|}{\sqrt{(x_4 - x_2)^2 + x_1^2}},$$

$$F = \frac{\pi \times P \times x_3^2}{4},$$

$$L_1 = \sqrt{(x_4 - x_2)^2 + x_1^2},$$

$$L_2 = \sqrt{(x_4 \times \sin \theta + x_1)^2 + (x_2 - x_4 \times \cos \theta)^2},$$

$$\theta = 45^\circ.$$

$$Q = 10000 \text{ lbs},$$

$$L = 240 \text{ in},$$

$$M_{\max} = 1.8 \times 10^6 \text{ lbs in},$$

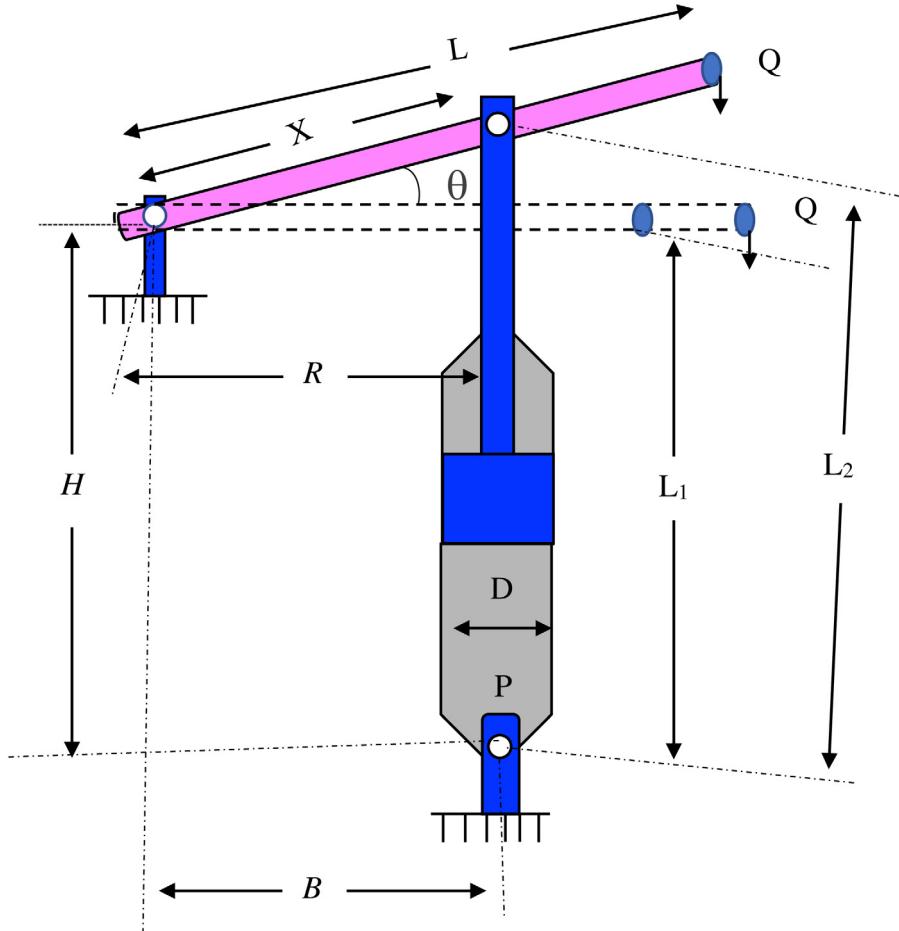
$$P = 1500 \text{ psi},$$

*Variable range:*

$$0.05 \leq x_1, x_2, x_4 \leq 500,$$

$$0.05 \leq x_3 \leq 120.$$

Moreover, the best solutions obtained by YDSE are presented in [Table 35](#). In addition, [Table 36](#) summarizes a comparison among the algorithms. According to the simulated results, YDSE outperforms its rivals in finding the minimum volume of the oil. For the best, mean, and worst values, YDSE gets a minimum volume of 8.4127E+00 and small SD equal to 6.9128E-13. We can see that the performance of the other algorithms is poor and has SD values. Also, the proposed algorithm takes a CPU time of 1.42 s, which is smaller than all the other algorithms except for HGS.

**Fig. 40.** The piston lever problem.**Table 36**

Results of the algorithms for the piston lever design problem.

Algorithm	Worst	Mean	Best	SD	CPU time
AVOA	1.6747E+02	1.9017E+01	8.4127E+00	4.0355E+01	1.65
CHIO	3.1717E+02	4.4790E+01	8.4863E+00	7.8733E+01	2.24
GTO	8.4128E+00	8.4127E+00	8.4127E+00	1.5370E-05	3.07
RUN	8.4127E+00	8.4127E+00	8.4127E+00	4.9266E-06	4.83
PSO	1.6747E+02	1.2506E+02	8.4127E+00	7.1541E+01	3.10
SMA	8.4127E+00	8.4127E+00	8.4127E+00	3.9541E-06	2.05
HGS	1.6747E+02	5.7719E+01	8.4127E+00	7.3587E+01	1.38
WOA	1.9110E+02	5.2088E+01	8.4802E+00	7.1304E+01	1.59
WSO	1.6747E+02	5.4574E+01	8.4127E+00	7.0297E+01	1.48
RSO	2.6608E+05	4.6251E+04	2.4225E+02	8.1991E+04	1.52
SCA	9.2957E+00	8.8570E+00	8.4979E+00	1.9111E-01	1.68
<b>YDSE</b>	<b>8.4127E+00</b>	<b>8.4127E+00</b>	<b>8.4127E+00</b>	<b>6.9128E-13</b>	<b>1.42</b>

#### 6.8.10. Welded beam design problem

In this problem, we need to optimize the values of four decision variables that represent weld thickness  $t(=x_1)$ , height  $h(=x_2)$ , length  $l(=x_3)$  and bar thickness  $b(=x_4)$ . The goal of the problem is to lessen the production costs, while respecting seven constraints of deflection, stress, welding, and geometry. Fig. 41 illustrates the structure of

the welded beam design problem. We can formulate the problem as:

*Minimize:*

$$f(x) = 1.10471 \times x_1^2 \cdot x_2 + 0.04811 \times x_3 \cdot x_4(14 + x_2),$$

*Subject to:*

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

$$g_2(x) = \delta(x) - \delta_{\max} \leq 0,$$

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

$$g_4(x) = x_1 - x_4 \leq 0,$$

$$g_5(x) = P - P_c(x) \leq 0,$$

$$g_6(x) = 0.125 - x_1 \leq 0,$$

$$g_7(x) = 1.10471 \times x_1^2 + 0.04811 \times x_3 \cdot x_4(14 + x_2) - 5 \leq 0,$$

$$\tau(x) = \sqrt{(\tau')^2 + 2 \times \tau' \times \tau'' \times \frac{x_2}{2R} + (\tau'')^2},$$

$$\tau' = \frac{P}{\sqrt{2}x_1 \cdot x_2},$$

$$\tau'' = \frac{MP}{J},$$

$$M = P(L + \frac{x_2}{2}),$$

$$R = \sqrt{\frac{x_2^2}{4} + (\frac{x_1 + x_3}{2})},$$

$$\delta(x) = \frac{6PL^3}{x_4 \cdot x_3^2},$$

$$\delta(x) = \frac{6PL^3}{Ex_4 \cdot x_3^2},$$

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

$$P = 6000 \text{ lb},$$

$$L = 14 \text{ in},$$

$$\delta_{\max} = 0.25 \text{ in},$$

$$E = 30 \times 10^6 \text{ psi},$$

$$G = 12 \times 10^6 \text{ psi},$$

$$\tau_{\max} = 13600 \text{ psi},$$

$$\delta_{\max} = 30000 \text{ psi},$$

*Variable range:*

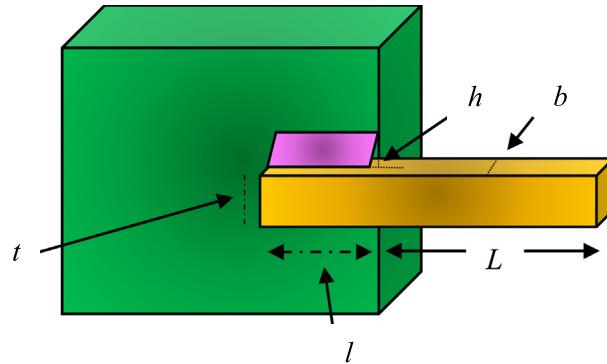
$$0.1 \leq x_1,$$

$$x_4 \leq 2,$$

$$0.1 \leq x_2,$$

$$x_3 \leq 10.$$

(36)



**Fig. 41.** Schematic of the welded beam structure with indication of design variables.

**Table 37**

The best results obtained by YDSE on the welded beam design problem.

	Symbols	Value
Variable	$x_1$	2.0573E-01
	$x_2$	3.4705E+00
	$x_3$	9.0366E+00
	$x_4$	2.0573E-01
Constraints	$g_1(x)$	0
	$g_2(x)$	-7.2760E-12
	$g_3(x)$	-3.0531E-16
	$g_4(x)$	-3.3907
	$g_5(x)$	-0.0807
	$g_6(x)$	-0.2355
	$g_7(x)$	-3.1832E-11
Objective function	$f(x)$	3.0881E-13

**Table 38**

Results of the algorithms for welded beam design problem.

Algorithm	Worst	Mean	Best	SD	CPU time
AVOA	1.8071E+00	1.7342E+00	1.7249E+00	1.9263E-02	1.51
CHIO	2.5382E+00	2.0600E+00	1.7890E+00	1.7792E-01	2.28
GTO	1.7286E+00	1.7252E+00	1.7249E+00	8.6801E-04	3.05
RUN	1.8316E+00	1.7567E+00	1.7249E+00	3.2376E-02	4.41
PSO	1.8143E+00	1.7278E+00	1.7249E+00	1.6330E-02	3.32
SMA	1.7256E+00	1.7251E+00	1.7249E+00	1.9738E-04	1.90
HGS	1.8023E+00	1.7322E+00	1.7249E+00	1.6414E-02	1.45
WOA	2.3433E+00	1.9119E+00	1.7623E+00	1.2659E-01	1.45
WSO	1.8143E+00	1.7278E+00	1.7249E+00	1.6330E-02	1.52
RSO	6.8689E+00	3.5386E+00	2.0761E+00	1.3027E+00	1.53
SCA	1.8637E+00	1.8242E+00	1.7610E+00	2.1615E-02	1.62
<b>YDSE</b>	<b>1.7249E+00</b>	<b>1.7249E+00</b>	<b>1.7249E+00</b>	<b>3.0881E-13</b>	<b>1.35</b>

The best values of the optimized decision variables and the constraints are presented in [Table 37](#). Also, the statistical results of the minimum production cost values for the algorithms are recorded in [Table 38](#) in terms of best, mean, worst, and SD. From the results, we can conclude that YDSE beats all the other competitors by attaining a minimum production cost of 1.7249E+00. Furthermore, it achieves the smallest SD value. Additionally, YDSE takes 1.35 s to solve the problem.

From the previous experiments on various kinds of functions (unimodal, multimodal, hybrid, composite, constrained and unconstrained) to investigate the performance of the YDSE optimizer, YDSE achieves superior

results in comparison with other well-regarded MH algorithms. In the exploration phase, if the solution has an odd number, it moves in the dark zones moving towards the central zone, which are expected to hold the best solution to skip from the local optima. During the exploitation phase, the algorithm exploits the prospective areas around the light regions, which are expected to hold the optimal solution. Several statistical analyses are employed, including average, standard deviation, visualized boxplots, CPU times and Wilcoxon signed rank-sum test. From the statistical analyses, we can conclude the following advantages. The YDSE optimizer is an amazing competitor to the other well-regarded MH algorithms and CEC winners of CEC 2014 and CEC 2017. It can balance the exploration and the exploitation phases effectively as can be inferred from the experiments. Also, it is efficient in terms of the CPU time compared to most of the algorithms. Regardless of the algorithm's efficiency in terms of the CPU time, some methods, such as the whale optimization algorithm, take less time. Furthermore, YDSE contains many parameters which can take much time to tune for each test function to find the best values for these parameters. Nevertheless, to avoid this issue, we previously tested the optimal values for these parameters, which remained consistent throughout the experiments.

## 7. Conclusions and future work

This study proposes a new physical MH algorithm inspired by the experiment of Young's double-slit that proves the wave nature of the light. According to the experiment, the outgoing waves from the two slits can interfere constructively or destructively resulting in bright and dark fringes on the projection screen. Many concepts are modeled from the experiment, including monochromatic light waves, Huygens' principle, constructive and destructive interference, wave intensity, amplitude, and path difference. Each fringe can be viewed as a possible solution. During the optimization process, the solution progresses across the search space based on the order number it possesses. In the exploration phase, if the solution has an odd number, it progresses through the dark sections towards the central zone, which is considered to hold the best solution to skip from the local optima. For the exploitation phase, the algorithm exploits the promising areas around the bright areas, which are assumed to contain the optimum. The YDSE optimizer is compared with many of the existing MHs on CEC 2014, CEC 2017, and CEC 2022 benchmarks. Ten engineering optimization problems are used for further investigating the performance of the algorithms. YDSE proved its superiority over the CEC 2014 and CEC 2017 winners, such as L-SHADE, LSHADE-cnEpSin and LSHADE-SPACMA. Several statistical analyses are employed, including average, standard deviation, visualized boxplots, CPU times and Wilcoxon signed rank-sum test. These analyses demonstrate the success of the proposed YDSE optimizer over their competitors. From the results, YDSE optimizer can be a strong competitor for solving the CEC competitions. However, the superior performance of the proposed algorithm, it has many parameters to configure. The parameter tuning is a time-consuming process that influences the algorithm performance. Thus, to save time, we choose the best parameter value by running different test functions and applications in this paper. However, there is still one limitation: if the values of these parameters are adjusted in different applications, the results may be improved. The following are some suggestions for the researchers to consider in the future. The incorporation of some stochastic operators can speed up the convergence of YDSE. The YDSE optimizer can be applied for tackling multi-objective optimization problems and many real-world applications.

## 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.

## Funding

This research has no funding source.

## Ethical approval

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

## Data availability

The code used in this paper can be obtained from this publicly accessible platform: [https://www.researchgate.net/publication/364243097\\_YDSE\\_Codes](https://www.researchgate.net/publication/364243097_YDSE_Codes) (accessed on 8 October 2022).

## References

- [1] K. Tafakkori, R. Tavakkoli-Moghaddam, A. Siadat, Sustainable negotiation-based nesting and scheduling in additive manufacturing systems: A case study and multi-objective meta-heuristic algorithms, *Eng. Appl. Artif. Intell.* 112 (2022) 104836.
- [2] S. Jain, D. Ramesh, D. Bhattacharya, A multi-objective algorithm for crop pattern optimization in agriculture, *Appl. Soft Comput.* 112 (2021) 107772.
- [3] G. Divsalar, et al., An optimization approach for green tourist trip design, *Soft Comput.* 26 (9) (2022) 4303–4332.
- [4] S. Ajayan, A.I. Selvakumar, Metaheuristic optimization techniques to design solar-fuel cell-battery energy system for locomotives, *Int. J. Hydrogen Energy* 47 (3) (2022) 1845–1862.
- [5] M. Cavazzuti, Deterministic optimization, in: *Optimization Methods*, Springer, 2013, pp. 77–102.
- [6] J. Chen, et al., Robust standard gradient descent algorithm for ARX models using aitken acceleration technique, *IEEE Trans. Cybern.* (2021).
- [7] P.E. Gill, W. Murray, Quasi-Newton methods for unconstrained optimization, *IMA J. Appl. Math.* 9 (1) (1972) 91–108.
- [8] J.A. Nelder, R. Mead, A simplex method for function minimization, *Comput. J.* 7 (4) (1965) 308–313.
- [9] J.J. Moré, The Levenberg–Marquardt algorithm: implementation and theory, in: *Numerical Analysis*, Springer, 1978, pp. 105–116.
- [10] C.-L. Chen, S.-C. Wang, Branch-and-bound scheduling for thermal generating units, *IEEE Trans. Energy Convers.* 8 (2) (1993) 184–189.
- [11] G. McMahon, P. Burton, Flow-shop scheduling with the branch-and-bound method, *Oper. Res.* 15 (3) (1967) 473–481.
- [12] J. Hu, et al., Dispersed foraging slime mould algorithm: Continuous and binary variants for global optimization and wrapper-based feature selection, *Knowl.-Based Syst.* 237 (2022) 107761.
- [13] M. Abdel-Basset, et al., A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection, *Expert Syst. Appl.* 139 (2020) 112824.
- [14] R.K. Yadav, R.P. Mahapatra, Hybrid metaheuristic algorithm for optimal cluster head selection in wireless sensor network, *Pervasive Mob. Comput.* 79 (2022) 101504.
- [15] Y. Feng, G.-G. Wang, A binary moth search algorithm based on self-learning for multidimensional knapsack problems, *Future Gener. Comput. Syst.* 126 (2022) 48–64.
- [16] F.S. Gharehchopogh, B. Abdollahzadeh, An efficient harris hawk optimization algorithm for solving the travelling salesman problem, *Cluster Comput.* 25 (3) (2022) 1981–2005.
- [17] T.S. Naseri, F.S. Gharehchopogh, A feature selection based on the farmland fertility algorithm for improved intrusion detection systems, *J. Netw. Syst. Manage.* 30 (3) (2022) 1–27.
- [18] S.E. Shukri, et al., Enhanced multi-verse optimizer for task scheduling in cloud computing environments, *Expert Syst. Appl.* 168 (2021) 114230.
- [19] L. Abualigah, et al., Aquila optimizer: a novel meta-heuristic optimization algorithm, *Comput. Ind. Eng.* 157 (2021) 107250.
- [20] H. Zamani, M.H. Nadimi-Shahroki, A.H. Gandomi, Starling murmuration optimizer: A novel bio-inspired algorithm for global and engineering optimization, *Comput. Methods Appl. Mech. Engrg.* 392 (2022) 114616.
- [21] A. Gogna, A. Tayal, Metaheuristics: review and application, *J. Exp. Theor. Artif. Intell.* 25 (4) (2013) 503–526.
- [22] R. Storn, K. Price, Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces, *J. Global Optim.* 11 (4) (1997) 341–359.
- [23] M.H. Nadimi-Shahroki, et al., MTDE: An effective multi-trial vector-based differential evolution algorithm and its applications for engineering design problems, *Appl. Soft Comput.* 97 (2020) 106761.
- [24] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, *IEEE Trans. Evol. Comput.* 3 (2) (1999) 82–102.
- [25] X. Yao, et al., Fast evolutionary algorithms, in: *Advances in Evolutionary Computing*, Springer, 2003, pp. 45–94.
- [26] H. Mühlenbein, G. Paaß, From recombination of genes to the estimation of distributions I. Binary parameters, in: *International Conference on Parallel Problem Solving from Nature*, Springer, 1996.
- [27] D. Simon, Biogeography-based optimization, *IEEE Trans. Evol. Comput.* 12 (6) (2008) 702–713.
- [28] P. Moscato, On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms, in: *Caltech Concurrent Computation Program, C3P Report*, 826, 1989, p. 1989.
- [29] D. Tang, et al., ITGO: Invasive tumor growth optimization algorithm, *Appl. Soft Comput.* 36 (2015) 670–698.
- [30] R. Eberhart, J. Kennedy, Particle swarm optimization, in: *Proceedings of the IEEE International Conference on Neural Networks*, Citeseer, 1995.
- [31] X. Xia, et al., Triple archives particle swarm optimization, *IEEE Trans. Cybern.* 50 (12) (2019) 4862–4875.
- [32] C. Charin, et al., A hybrid of bio-inspired algorithm based on levy flight and particle swarm optimizations for photovoltaic system under partial shading conditions, *Sol. Energy* 217 (2021) 1–14.
- [33] J. Liang, et al., Classified perturbation mutation based particle swarm optimization algorithm for parameters extraction of photovoltaic models, *Energy Convers. Manage.* 203 (2020) 112138.
- [34] D. Karaboga, B. Basturk, Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems, in: *International Fuzzy Systems Association World Congress*, Springer, 2007.

- [35] S.S. Jadon, et al., Artificial bee colony algorithm with global and local neighborhoods, *Int. J. Syst. Assur. Eng. Manag.* 9 (3) (2018) 589–601.
- [36] F. Zhong, H. Li, S. Zhong, A modified ABC algorithm based on improved-global-best-guided approach and adaptive-limit strategy for global optimization, *Appl. Soft Comput.* 46 (2016) 469–486.
- [37] S.S. Jadon, et al., Accelerating artificial bee colony algorithm with adaptive local search, *Memet. Comput.* 7 (3) (2015) 215–230.
- [38] X.-S. Yang, S. Deb, Cuckoo search via Lévy flights, in: 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), Ieee, 2009.
- [39] G. Dhiman, et al., A novel algorithm for global optimization: rat swarm optimizer, *J. Ambient Intell. Humaniz. Comput.* 12 (8) (2021) 8457–8482.
- [40] M. Braik, et al., White Shark Optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems, *Knowl.-Based Syst.* 243 (2022) 108457.
- [41] B. Abdollahzadeh, F.S. Gharehchopogh, S. Mirjalili, African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems, *Comput. Ind. Eng.* 158 (2021) 107408.
- [42] S. Mirjalili, A. Lewis, The whale optimization algorithm, *Adv. Eng. Softw.* 95 (2016) 51–67.
- [43] C. Zhong, G. Li, Z. Meng, Beluga whale optimization: A novel nature-inspired metaheuristic algorithm, *Knowl.-Based Syst.* (2022) 109215.
- [44] A. Faramarzi, et al., Marine Predators Algorithm: A nature-inspired metaheuristic, *Expert Syst. Appl.* 152 (2020) 113377.
- [45] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Softw.* 69 (2014) 46–61.
- [46] X.-S. Yang, Firefly algorithm, levy flights and global optimization, in: Research and Development in Intelligent Systems XXVI, Springer, 2010, pp. 209–218.
- [47] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, *Knowl.-Based Syst.* 89 (2015) 228–249.
- [48] S. Mirjalili, et al., Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems, *Adv. Eng. Softw.* 114 (2017) 163–191.
- [49] A. Mohammadi-Balani, et al., Golden eagle optimizer: A nature-inspired metaheuristic algorithm, *Comput. Ind. Eng.* 152 (2021) 107050.
- [50] M. Jain, V. Singh, A. Rani, A novel nature-inspired algorithm for optimization: Squirrel search algorithm, *Swarm Evol. Comput.* 44 (2019) 148–175.
- [51] H.R.R. Zaman, F.S. Gharehchopogh, An improved particle swarm optimization with backtracking search optimization algorithm for solving continuous optimization problems, *Eng. Comput.* (2021) 1–35.
- [52] M.J. Goldanloo, F.S. Gharehchopogh, A hybrid OBL-based firefly algorithm with symbiotic organisms search algorithm for solving continuous optimization problems, *J. Supercomput.* 78 (3) (2022) 3998–4031.
- [53] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: a gravitational search algorithm, *Inform. Sci.* 179 (13) (2009) 2232–2248.
- [54] A. Faramarzi, et al., Equilibrium optimizer: A novel optimization algorithm, *Knowl.-Based Syst.* 191 (2020) 105190.
- [55] S. Li, et al., Slime mould algorithm: A new method for stochastic optimization, *Future Gener. Comput. Syst.* 111 (2020) 300–323.
- [56] I. Ahmadianfar, et al., RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method, *Expert Syst. Appl.* 181 (2021) 115079.
- [57] S. Mirjalili, SCA: a sine cosine algorithm for solving optimization problems, *Knowl.-Based Syst.* 96 (2016) 120–133.
- [58] L. Abualigah, et al., The arithmetic optimization algorithm, *Comput. Methods Appl. Mech. Engrg.* 376 (2021) 113609.
- [59] O.K. Erol, I. Eksin, A new optimization method: big bang–big crunch, *Adv. Eng. Softw.* 37 (2) (2006) 106–111.
- [60] J.L.J. Pereira, et al., Lichtenberg algorithm: A novel hybrid physics-based meta-heuristic for global optimization, *Expert Syst. Appl.* 170 (2021) 114522.
- [61] S. Mirjalili, S.M. Mirjalili, A. Hatamlou, Multi-verse optimizer: a nature-inspired algorithm for global optimization, *Neural Comput. Appl.* 27 (2) (2016) 495–513.
- [62] Y. Zhang, Z. Jin, S. Mirjalili, Generalized normal distribution optimization and its applications in parameter extraction of photovoltaic models, *Energy Convers. Manage.* 224 (2020) 113301.
- [63] R.V. Rao, V.J. Savsani, D. Vakharia, Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems, *Comput. Aided Des.* 43 (3) (2011) 303–315.
- [64] F. Ramezani, S. Lotfi, Social-based algorithm (SBA), *Appl. Soft Comput.* 13 (5) (2013) 2837–2856.
- [65] S.H.S. Moosavi, V.K. Bardsiri, Poor and rich optimization algorithm: A new human-based and multi populations algorithm, *Eng. Appl. Artif. Intell.* 86 (2019) 165–181.
- [66] S. Satapathy, A. Naik, Social group optimization (SGO): a new population evolutionary optimization technique, *Complex Intell. Syst.* 2 (3) (2016) 173–203.
- [67] S.Q. Salih, A.A. Alsewari, A new algorithm for normal and large-scale optimization problems: Nomadic People Optimizer, *Neural Comput. Appl.* 32 (14) (2020) 10359–10386.
- [68] N. Moosavian, B.K. Roodsari, Soccer league competition algorithm: A novel meta-heuristic algorithm for optimal design of water distribution networks, *Swarm Evol. Comput.* 17 (2014) 14–24.
- [69] A.H. Kashan, League Championship Algorithm (LCA): An algorithm for global optimization inspired by sport championships, *Appl. Soft Comput.* 16 (2014) 171–200.
- [70] N. Razmjoo, M. Khalilpour, M. Ramezani, A new meta-heuristic optimization algorithm inspired by FIFA world cup competitions: theory and its application in PID designing for AVR system, *J. Control Autom. Electr. Syst.* 27 (4) (2016) 419–440.
- [71] P.R. Singh, M. Abd Elaziz, S. Xiong, Ludo game-based metaheuristics for global and engineering optimization, *Appl. Soft Comput.* 84 (2019) 105723.

- [72] E. Fadakar, M. Ebrahimi, A new metaheuristic football game inspired algorithm, in: 2016 1st Conference on Swarm Intelligence and Evolutionary Computation, CSIEC, IEEE, 2016.
- [73] W. Lv, et al., Election campaign algorithm, in: 2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR 2010), IEEE, 2010.
- [74] Q. Askari, I. Younas, M. Saeed, Political Optimizer: A novel socio-inspired meta-heuristic for global optimization, *Knowl.-Based Syst.* 195 (2020) 105709.
- [75] J.L. Melvix, Greedy politics optimization: Metaheuristic inspired by political strategies adopted during state assembly elections, in: 2014 IEEE International Advance Computing Conference, IACC, IEEE, 2014.
- [76] E. Atashpaz-Gargari, C. Lucas, Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition, in: 2007 IEEE Congress on Evolutionary Computation, Ieee, 2007.
- [77] A. Borji, A new global optimization algorithm inspired by parliamentary political competitions, in: Mexican International Conference on Artificial Intelligence, Springer, 2007.
- [78] H. Shayanfar, F.S. Gharehchopogh, Farmland fertility: A new metaheuristic algorithm for solving continuous optimization problems, *Appl. Soft Comput.* 71 (2018) 728–746.
- [79] M.A. Al-Betar, et al., Coronavirus herd immunity optimizer (CHIO), *Neural Comput. Appl.* 33 (10) (2021) 5011–5042.
- [80] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, *IEEE Trans. Evol. Comput.* 1 (1) (1997) 67–82.
- [81] O. Donati, G. Missiroli, G. Pozzi, An experiment on electron interference, *Amer. J. Phys.* 41 (5) (1973) 639–644.
- [82] M. Kenmoku, K. Kume, Young's double slit experiment in quantum field theory, 2011, arXiv preprint arXiv:1103.0100.
- [83] W. Rueckner, J. Peidle, Young's double-slit experiment with single photons and quantum eraser, *Amer. J. Phys.* 81 (12) (2013) 951–958.
- [84] I. Newton, A new theory about light and colors, *Amer. J. Phys.* 61 (2) (1993) 108–112.
- [85] M.S. Zubairy, A very brief history of light, in: Optics in Our Time, Springer, Cham, 2016, pp. 3–24.
- [86] B.B. Baker, E.T. Copson, The Mathematical Theory of Huygens' Principle. Vol. 329, American Mathematical Soc, 2003.
- [87] G. Paul, Huygens' Principle and Hyperbolic Equations, Academic Press, 2014.
- [88] A. Aspect, From Huygens' waves to Einstein's photons: Weird light, *C. R. Phys.* 18 (9–10) (2017) 498–503.
- [89] T. Young, I. The Bakerian Lecture, Experiments and calculations relative to physical optics, *Philos. Trans. R. Soc. Lond.* (94) (1804) 1–16.
- [90] A.L. Schawlow, Lasers: the intense, monochromatic, coherent light from these new sources shows many unfamiliar properties, *Science* 149 (3679) (1965) 13–22.
- [91] L. Learning, Young's double slit experiment, in: Fundamentals of Heat, Light & Sound, 2021.
- [92] R.A. Serway, J.W. Jewett, Physics for Scientists and Engineers, Cengage learning, 2018.
- [93] H.D. Young, et al., Sears and Zemansky's University Physics, Pearson, 2014.
- [94] M. Born, E. Wolf, Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light, Elsevier, 2013.
- [95] J.I. Thomas, The classical double slit interference experiment: a new geometrical approach, *Am. J. Opt. Photonics* 7 (1) (2019) 1–9.
- [96] D. Sang, G. Jones, Cambridge International AS and A Level Physics Workbook with CD-ROM, Cambridge University Press, 2016.
- [97] P.A. Tipler, G. Mosca, Physics for Scientists and Engineers, Macmillan, 2007.
- [98] J.J. Liang, B.Y. Qu, P.N. Suganthan, Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization, Vol. 635, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, 2013, p. 490.
- [99] G. Wu, R. Mallipeddi, P.N. Suganthan, Problem Definitions and Evaluation Criteria for the CEC 2017 Competition on Constrained Real-Parameter Optimization, Technical Report, National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, 2017.
- [100] W. Luo, et al., Benchmark functions for CEC 2022 competition on seeking multiple optima in dynamic environments, 2022, arXiv preprint arXiv:2201.00523.
- [101] M. Abd El Aziz, et al., Prediction of biochar yield using adaptive neuro-fuzzy inference system with particle swarm optimization, in: 2017 IEEE PES PowerAfrica, IEEE, 2017.
- [102] P. Mehta, et al., Hunger games search algorithm for global optimization of engineering design problems, *Mater. Test.* 64 (4) (2022) 524–532.
- [103] T.S. Ayyarao, et al., War strategy optimization algorithm: a new effective metaheuristic algorithm for global optimization, *IEEE Access* 10 (2022) 25073–25105.
- [104] I. Ahmadianfar, O. Bozorg-Haddad, X. Chu, Gradient-based optimizer: A new metaheuristic optimization algorithm, *Inform. Sci.* 540 (2020) 131–159.
- [105] R. Poláková, J. Tvrdík, P. Bujok, Evaluating the performance of L-SHADE with competing strategies on CEC2014 single parameter-operator test suite, in: 2016 IEEE Congress on Evolutionary Computation, CEC, IEEE, 2016.
- [106] N.H. Awad, M.Z. Ali, P.N. Suganthan, Ensemble sinusoidal differential covariance matrix adaptation with euclidean neighborhood for solving CEC2017 benchmark problems, in: 2017 IEEE Congress on Evolutionary Computation, CEC, IEEE, 2017.
- [107] A.W. Mohamed, et al., 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.
- [108] H. Bayzidi, et al., Social network search for solving engineering optimization problems, *Comput. Intell. Neurosci.* 2021 (2021).
- [109] B. Abdollahzadeh, F. Soleimanian Gharehchopogh, S. Mirjalili, Artificial gorilla troops optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems, *Int. J. Intell. Syst.* 36 (10) (2021) 5887–5958.
- [110] J.-F. Tsai, et al., Optimization Theory, Methods, and Applications in Engineering 2013, Hindawi, 2014.