



Kepler optimization algorithm: A new metaheuristic algorithm inspired by Kepler's laws of planetary motion



Mohamed Abdel-Basset^a, Reda Mohamed^a, Shaimaa A. Abdel Azeem^a,
Mohammed Jameel^b, Mohamed Abouhawwash^{c,d,*}

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

^b Department of Mathematics, Faculty of Science, Sana'a University, Sana'a 13509, Yemen

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

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

ARTICLE INFO

Article history:

Received 14 January 2023

Received in revised form 18 February 2023

Accepted 7 March 2023

Available online 11 March 2023

Keywords:

Kepler's laws

Metaheuristics

Optimization

Constrained problems

Photovoltaic modules

ABSTRACT

This study presents a novel physics-based metaheuristic algorithm called Kepler optimization algorithm (KOA), inspired by Kepler's laws of planetary motion to predict the position and velocity of planets at any given time. In KOA, each planet with its position acts as a candidate solution, which is randomly updated through the optimization process with respect to the best-so-far solution (Sun). KOA allows for a more effective exploration and exploitation of the search space because the candidate solutions (planets) exhibit different situations from the Sun at different times. Four challengeable benchmarks, namely CEC 2014, CEC 2017, CEC 2020, and CEC2022, and eight constrained engineering design problems, in addition to the parameter estimation problem of photovoltaic modules, were used to assess the performance of KOA. To observe its effectiveness, it was compared with three classes of stochastic optimization algorithms, including: (i) the latest published algorithms, including Snake Optimizer (SO), Fick's Law Algorithm (FLA), Coati Optimization Algorithm (COA), Pelican Optimization Algorithm (POA), Dandelion Optimizer (DO), Mountain Gazelle Optimizer (MGO), Artificial Gorilla Troops Optimizer (GTO), and Slime Mold Algorithm (SMA); (ii) well-studied and highly cited algorithms, such as Whale Optimization Algorithm (WOA) and Grey Wolf Optimizer (GWO); and (iii) two highly performing optimizers: LSHADE-cnEpSin and LSHADE-SPACMA. Results of the convergence curve and statistical information indicated that KOA is more promising than all the compared optimizers. The source code of KOA is publicly accessible at <https://www.mathworks.com/matlabcentral/fileexchange/126175-kepler-optimization-algorithm-koa>

Published by Elsevier B.V.

1. Introduction

Several real-world optimization problems in the fields of machine learning and artificial intelligence can be categorized as continuous, discrete, constrained, or unconstrained [1]. Conventional mathematical programming methods, such as conjugate gradient, fast steepest, quasi-Newton methods, and sequential quadratic programming, may be challenging to apply when attempting to solve certain classes of these optimization problems because of the characteristics described above [2,3]. Some studies have shown that these methods are not as efficient as they should

be or not always efficient when dealing with many larger-scale real-world multimodal, discrete, and non-differentiable optimization problems [4,5]. Therefore, metaheuristic algorithms have been developed and used as competitive alternative approaches for handling several optimization problems with different characteristics, because these algorithms are relatively straightforward and easy to implement [6–8]. The primary advantage of these techniques is that they do not require knowing the gradient or mathematical characteristics of the objective landscape. Nevertheless, the majority of metaheuristic algorithms share a common flaw, i.e., they frequently exhibit a delicate sensitivity to the tuning of controlling parameters. Other disadvantages of these methods include the likelihood of suffering from premature convergence, local optimum entrapment, and lack of population diversity [9].

Metaheuristic algorithms are divided into two major categories [10]: population- and single solution-based algorithms. In the latter, only one solution is considered throughout the

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

E-mail addresses: mohamedbasset@ieee.org (M. Abdel-Basset), redamoh@zu.edu.eg (R. Mohamed), shaimaaayman204@gmail.com (S.A.A. Azeem), moh.jameel@su.edu.ye (M. Jameel), abouhaww@msu.edu, saleh1284@mans.edu.eg (M. Abouhawwash).

optimization stage; in the former, a population of solutions is evolved over each iteration. Population-based metaheuristic algorithms (P-metaheuristics) can find optimal or near-optimal solutions for a variety of optimization issues. These algorithms are useful for avoiding local optima because several solutions support one another and achieve good exploration of the search space. They also have the ability to jump to a promising area of the search space. These P-metaheuristic algorithms begin their search by distributing a population of individuals within the search space, with each individual consisting of d-dimensions to create a vector representing the solution to the tackled optimization problem. P-metaheuristics can be classified into four categories: evolution-, swarm-, physics-, and human-based algorithms. Before going deeply in details of each category, let us illustrate some optimization problems tackled by these algorithms.

P-metaheuristics could be applied to tackle several optimization problems, which could be classified to discrete or continuous [11]. The continuous problems are compounded of decimal decision variables that have to be optimized to find the near optimal solutions; while the discrete problems consist of decision variables that might be represented in binary or integer according to the tackled problem and representation schema [12]. These optimization problems may have a single, multiple, or many objectives. The single-objective problems could be directly tackled by the P-metaheuristics, because they involve a single objective value that needs to be maximized or minimized according to the needs of this problem. On the contrary, both multi- and many objectives necessitate the use of an additional theory, such as Pareto optimality, in order to handle all of the objectives in conflict simultaneously and be solvable by these algorithms [13]. Unfortunately, the P-metaheuristics in the classical form could not perform well for several optimization problems. Therefore, the researchers, over the last few decades, have strived to enhance their performance using various concepts listed below [14,15]:

- Hybridization between two or more P-metaheuristics.
- Quantum computing concept
- Opposition-based theory
- Chaotic maps
- Lévy-Flight strategy

However, both the traditional and modified P-metaheuristics have a number of problematic flaws, including a tendency to get stuck in local optima, a slow convergence speed, and a lack of population variety. Therefore, this study introduces the Kepler optimization algorithm (KOA) as a new and reliable physics-based metaheuristic method for solving the continuous optimization problems. KOA is based on Kepler's laws about how planets move. The position, mass, gravitational force, and orbital velocity of a planet are the four fundamental operators that govern the orbital path of planets around the Sun in accordance with these laws. These operators serve as the foundation for KOA. In KOA, planets, also known as candidate solutions, exhibit distinct relationships with the Sun at different periods, and thus, it allows for more effective exploration and exploitation of the search space. The performance of KOA was evaluated using four challengeable benchmarks of the Congress on Evolutionary Computation (CEC): CEC 2014, CEC 2017, CEC 2020, and CEC2022. In addition, eight constrained engineering design challenges and the parameter estimate problem of solar modules were utilized to reveal its ability to handle several real-world optimization problems. KOA was compared with three types of stochastic optimization techniques to determine its efficacy. The first class contains some of the most recently published algorithms. The second class comprises some of the most well-studied and highly cited algorithms. The third class includes high-performance optimizers. The convergence curves and statistical data demonstrated that KOA is

the most promising among the competing optimizers. The main contributions of this study are as follows:

- Proposing a novel physics-based metaheuristic algorithm, known as Kepler optimization algorithm (KOA), which is inspired by Kepler's laws of planetary motion.
- Evaluating KOA using four challengeable CEC benchmarks, eight constrained engineering design challenges, and the parameter estimate problem of solar modules to reveal its ability to handle numerous optimization problems with various characteristics.
- The experimental findings show that KOA performs better than three different classes of compared optimizers.

The remainder of this paper is organized as follows. Section 2 presents the literature review. Section 3 presents the inspiration of the proposed KOA. Section 4 explains the mathematical model and pseudocode of the proposed algorithm. Section 5 provides the results and discussion. Section 6 includes the results and discussion for solving nine constrained optimization problem by using KOA and its rival optimizers. Section 7 presents the results and discussion for solving the parameter estimation problem of photovoltaic modules. Finally, Section 8 draws the conclusions and offers suggestions for future work.

2. Related work

As mentioned above, P-metaheuristics are divided into four major categories: evolution-, swarm-, physics-, and human-based algorithms. The first category, called swarm intelligence algorithms, is inspired by the social behavior of plants, insects, birds, and other animals. For example, the flocking behavior of birds served as the primary source of inspiration for the particle swarm optimization (PSO) algorithm proposed by Eberhart and Kennedy [16]. Before the optimization process, PSO creates a number of particles to represent candidate solutions for an optimization problem. Thereafter, the optimization process is started to update each particle on the basis of its local best position and the global best particle. The whale optimization algorithm [17], dwarf mongoose optimization algorithm [18], grasshopper optimization algorithm [19], artificial gorilla troops optimizer [20], horse herd optimization algorithm [21], ant colony optimization [22], nutcracker optimizer [23], chameleon swarm algorithm [24], cuckoo search [25], honey badger algorithm [26], duck swarm algorithm [27], Aquila optimizer [28], grey wolf optimization [29], marine predators algorithm [30], artificial hummingbird algorithm [31], Harris hawks optimization [32], reptile search algorithm [33], and slime mold algorithm [34] are additional algorithms based on swarm intelligence behavior.

The second group of algorithms, known as physics-based algorithms, imitate physical laws, such as the force of inertia, electromagnetic force, and gravitational force [35]. Some examples of physics-based metaheuristic algorithms are the water cycle algorithm [36], gravitational search algorithm [37], simulated annealing [38], big-bang big-crunch [39], galaxy-based search algorithm [40], central force optimization [41], black hole algorithm [42], charged system search [43], ray optimization algorithm [44], curved space optimization [45], artificial physics algorithm [46], Henry gas solubility optimization algorithm [47], space gravitational algorithm [48], equilibrium optimizer [49], small-world optimization algorithm [50], artificial chemical reaction optimization algorithm [51], multiverse optimizer [52], electromagnetism-like algorithm [53], intelligent water drops algorithm [54], gravitational local search algorithm [55], ion motion algorithm [56], integrated radiation algorithm [57], river formation dynamics algorithm [58], Young's double-slit experiment

Table 1
Classification of some recently published P-metaheuristics.

Algorithm	Year	Inspiration	Classification	Reference
Orca Optimization Algorithm (OOA)	2020	Unique wave-washing orcas hunting technique	Swarm-based	[78]
Remora optimization algorithm (ROA)	2021	Parasitic behavior of remora	Swarm-based	[79]
Carnivorous Plant algorithms	2021	How the carnivorous plants adapting to survive in the harsh environment		[80]
Golden Eagle Optimizer (GEO)	2021	Intelligence of golden eagles in tuning speed	Swarm-based	[81]
Fire Hawk Optimizer (FHO)	2022	Foraging behavior of whistling kites, black kites and brown falcons	Evolutionary	[82]
Ebola Optimization Search Algorithm (EOSA)	2022	Propagation strategy of Ebola virus	Evolutionary	[83]
White Shark Optimizer (WSO)	2022	Behaviors of great white sharks	Swarm-based	[84]
Growth Optimizer (GO)	2022	Learning and reflection mechanisms of Individuals	Human-based	[85]
Pelican Optimization Algorithm (POA)	2022	Hunting behavior of pelicans	Swarm-based	[86]
Mountain Gazelle Optimizer (MGO)	2022	Gazelles' hierarchical and social life	Swarm-based	[87]
Snake optimizer (SO)	2022	Mating behavior of snakes	Swarm-based	[88]
Dandelion optimizer (DO)	2022	Dandelion seed long-distance flight	Swarm-based	[89]
Coati optimization algorithm (COA)	2023	Coati behavior in nature	Swarm-based	[90]
Termit life cycle optimizer (TLCO)	2023	Life cycle of a termite colony	Swarm-based	[91]
Fick's Law Algorithm (FLA)	2023	Fick's law	Physics-based	[92]
Running City game optimizer (RCGO)	2023	Running city game	Human-based	[93]
Artificial locust swarm optimization algorithm (ALSO))	2023	Behavior of locust swarms	Swarm-based	[94]
Siberian tiger optimization (STO)	2022	hunting and fighting behavior of Siberian tiger	Swarm-based	[95]
Special relativity search (SRS)	2022	Interaction of particles in an electromagnetic field	Physics-based	[96]
Artificial rabbits optimization (ARO)	2022	Survival strategies of rabbits	Swarm-based	[97]
Starling murmuration optimizer (SMO)	2022	Starlings' behaviors	Swarm-based	[98]

optimizer [59], light spectrum optimizer [60], and Archimedes optimization algorithm [61].

The third category of P-metaheuristics includes algorithms that mimic the biological evolutionary behavior of living things. Genetic algorithms that model the Darwinian theory of evolution through mutation, recombination, and natural selection were the first evolutionary algorithms (EAs) to be developed, and they continue to be the most popular type of EAs at present [62]. Other popular algorithms in this category are genetic programming [63], evolution strategy [64], evolutionary programming [65], differential evolution [66], biogeography-based optimizer [67], gradient evolution algorithm [68], backtracking search optimization algorithm [69], and tree-seed algorithm [70]. The fourth category consists of human-based algorithms, which are designed to mimic the social behavior of humans. The political optimizer [71], past present future [72], harmony search [73], teaching–learning-based optimization [74], exchange market algorithm [75], brain storm optimization [76], and soccer league competition [77] are common examples that belong to this category. Table 1 categorizes some recently published P-metaheuristics.

3. Inspiration

In the ancient past, the prevalent idea about the universe was that the Sun, planets, stars, and moons revolved around the Earth until the Polish astronomer Copernicus refuted this idea in 1543. Copernicus introduced heliocentric theory, i.e., that the Earth and the planets revolve around the Sun [99]. However, Copernicus did not have the tools to prove his theory. In 1609, the German astronomer Johannes Kepler finally proved heliocentric theory [100,101]. Kepler performed numerous complex calculations and introduced three basic laws that explain how planets move around the Sun. Kepler's laws of planetary motion are described as follows [100,102].

- **Kepler's First Law:** All planets move in elliptical orbits, with the Sun at one focus. This law describes the shape of the orbits in which planets move around the Sun. These orbits are elliptical, as shown in Fig. 1. An ellipse is similar to an oval, and thus, this shape has two focal points. The Sun lies at one of these primary focal points. The number that

characterizes how flat an ellipse looks is called eccentricity, which is denoted by e . Eccentricity e can be calculated by taking the center-to-focus distance and dividing it by the semimajor axis distance. The limiting cases are the circle ($e = 0$) and a line segment ($e = 1$). Fig. 2 depicts different ellipse shapes.

- **Kepler's Second Law:** A line that connects a planet to the Sun sweeps out equal areas at equal times. This law describes the velocity at which a planet moves when it revolves around the Sun; that is, the velocity at which a planet moves through space constantly changes. A planet moves faster when it is close to the Sun and slower when it is far from the Sun. However, if an imaginary line is drawn from the planet's center to the center of the Sun, then that line will sweep the same area at equal time intervals. The velocity of a planet around the Sun can be calculated using the vis-viva equation [103], which is given using Eq. (1):

$$V = \left[\mu(M_s + m) \left(\frac{2}{R} - \frac{1}{a} \right) \right]^{\frac{1}{2}}, \quad (1)$$

where M_s and m denote the mass of the Sun and a planet, respectively; R is the Euclidean distance between the Sun and a planet at this time; μ is the universal gravitational constant; and a is the semimajor axis of the orbit.

- **Kepler's Third Law:** For any planet, the square of the period of any planet is proportional to the cube of the semimajor axis of its orbit. This law can be mathematically expressed using Eq. (2):

$$T^2 = \left[\frac{4\pi^2}{\mu (M_s + m)} \right] a^3, \quad (2)$$

where T is the period of the orbiting body (the time required to complete one orbit), a is the semimajor axis of the orbit, μ is the universal gravitational constant, M_s is the mass of the Sun, and m is the mass of a planet.

Most of the objects in our solar system rotate counterclockwise. However, two objects in our solar system do not rotate in the same way as the rest of the planets. For example, Venus rotates about its axis in a clockwise direction. Such oddities are believed to be caused by events, such as collisions, which occurred during the formation of the solar system.

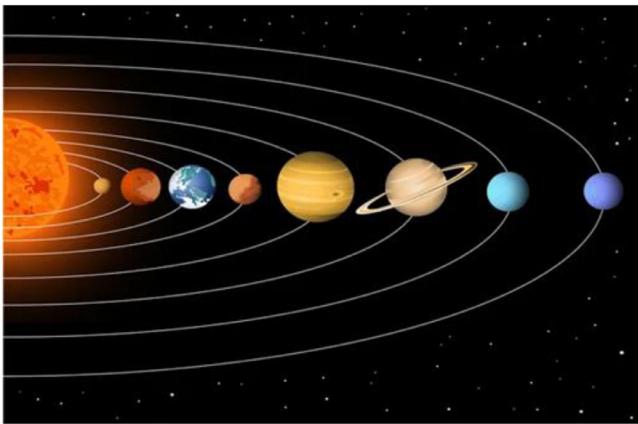


Fig. 1. The motion of a planet around the Sun in an elliptical orbit.

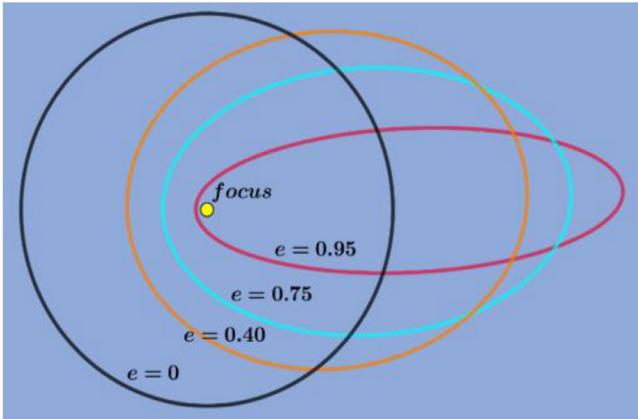


Fig. 2. Different cases of ellipse shapes.

In accordance with the aforementioned laws, four basic operators affect the path of planets around the Sun: the position, mass, gravitational force, and orbital velocity of a planet. These operators represent the basis for building the mathematical model of the proposed algorithm. Theoretically, Kepler's laws allow the prediction of the position and velocity of planets at any given time. This scenario represents the exact inspiration of KOA. The following section provides the mathematical model of KOA.

4. Kepler optimization algorithm (KOA)

This section presents a new optimization algorithm inspired by Kepler's laws of planetary motion [101]. In our proposed algorithm, Kepler's first law is simulated as follows. The Sun and the planets (or objects) revolving around it in (imaginary) elliptical orbits can be used to represent the search space, as shown in Fig. 3. In KOA, the planets (candidate solutions) are under different situations from the Sun (the best solution) at various times, and thus, the search space is explored and exploited more efficiently. Fig. 3 illustrates how the position of an object, its mass, the force of attraction between the object and the Sun, and the velocity with which it orbits the Sun change its position around the best solution, i.e., the Sun. The figure also depicts the rotation of most objects in a clockwise direction. Fig. 3 can also be used to explain how the searcher's position changes in 3D space.

Similar to other metaheuristic population-based algorithms, KOA starts the search process with an initial set of objects (candidate solutions) with stochastic orbitals. Each object is initialized

with its random position in orbit during this stage. After evaluating the fitness of the initial set, KOA runs in iterations until the termination condition is met. In the current study, we use the term "time" instead of "iteration" because it is a common term in solar system theory and cosmology.

During optimization, the following rules are applied to KOA.

- The orbital period of a planet (the candidate solution) is chosen randomly in accordance with the normal distribution.
- The eccentricity of a planet's orbit is selected at random from a range of 0 to 1.
- The fitness of a solution is calculated on the basis of the objective function.
- The best solution, in iteration, is the central star (the Sun).
- The distance between the Sun and the planet is changed in accordance with the current time.

The rest of this section presents the mathematical model of KOA. In brief, the pseudocode and flowchart of KOA are presented in Algorithm 1 and Fig. 6, respectively. The time complexity of the steps listed in this algorithm is of $O(NT_{max})$, where N represents the population size, and T_{max} is the termination criteria of the proposed KOA based on the maximum number of function evaluation. Theoretically, KOA can be considered a global optimization algorithm because it includes exploration and exploitation phases. Mathematically, the processes of the proposed KOA are described in detail as follows.

Step 1: Initialization process

In this process, a number of planets equal to N , referred to as the population size, will be randomly distributed in d -dimensions, representing the decision variables of an optimization problem, in accordance with the following formula:

$$X_i^j = X_{i,low}^j + rand_{[0,1]} \times (X_{i,up}^j - X_{i,low}^j), \begin{cases} i = 1, 2, \dots, N, \\ j = 1, 2, \dots, d. \end{cases} \quad (3)$$

where X_i indicates the i th planet (candidate solution) in the search space; N represents the number of solution candidates in search space; d represents the dimension of the problem to be optimized; $X_{i,up}^j$ and $X_{i,low}^j$ represent the upper and lower bounds, respectively, of the j th decision variable; and $rand_{[0,1]}$ is a number generated randomly between 0 and 1.

The orbital eccentricity (e) for each i th object is initialized using Eq. (4):

$$e_i = rand_{[0,1]}, i = 1, \dots, N; \quad (4)$$

where $rand_{[0,1]}$ is a random value generated within interval $[0, 1]$. Finally, the orbital period (T) for each i th object is initialized using Eq. (5):

$$T_i = |r|, i = 1, \dots, N; \quad (5)$$

where r is the number generated randomly on the basis of the normal distribution.

Step 2: Defining the gravitational force (F)

The Sun is the main element of the solar system; it represents the largest object in the solar system and controls the movement of the group through its gravity [104]. The primary reason why planets orbit the Sun is because the Sun's gravity keeps them in their orbit. If the Sun does not exist, then planets will move in a straight line toward infinity; however, the Sun's gravity constantly changes direction to enable planets to move around it in an elliptical shape [105]. Gravity is known as the fundamental force that controls the orbits of planets around the Sun. Each planet has its own gravity that is proportional to its size. Notably, the velocity of a planet depends on the gravity of the Sun. The

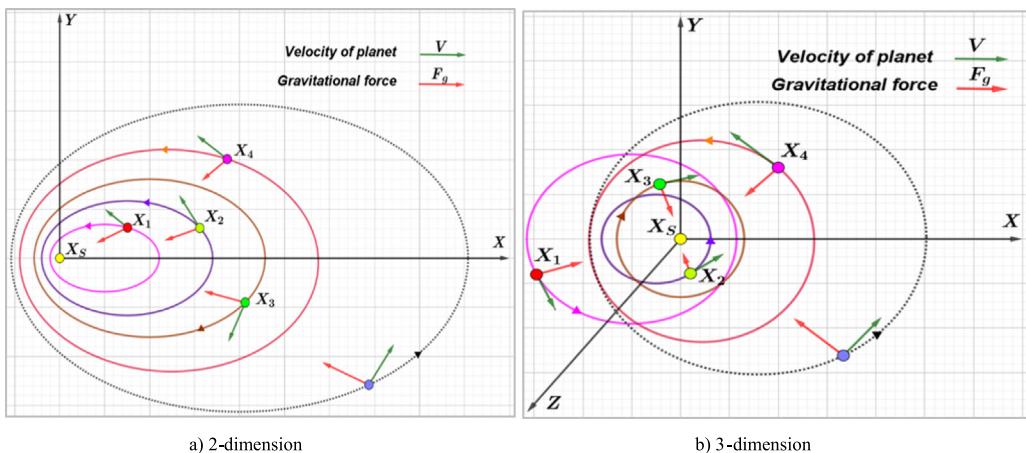


Fig. 3. Possible positions in 2-dimension and 3-dimension.

closer a planet is to the Sun, the greater its orbital velocity, and vice versa. The attraction force of the Sun X_S and any planet X_i is given by the universal law of gravitation [106], which is defined as

$$\mathbf{F}_{g_i}(t) = e_i \times \mu(t) \times \frac{\bar{M}_s \times \bar{m}_i}{\bar{R}_i^2 + \varepsilon} + r_1, \quad (6)$$

where \bar{M}_s and \bar{m}_i denote the normalized values of M_s and m_i , which represent the mass of X_s and X_i , respectively, and given by Eqs. (8) and (9); ε is a small value; μ is the universal gravitational constant; e_i is the eccentricity of a planet's orbit, which is a value between 0 and 1 that was proposed to endow a stochastic characteristic to KOA; r_1 is a value that is generated randomly between 0 and 1 to give more variation to the gravitation values within the optimization process; and \bar{R}_i is the normalized value of R_i that represents the Euclidian distance between X_s and X_i , and is defined as

$$R_i(t) = \|X_S(t) - X_i(t)\|_2 = \sqrt{\sum_{j=1}^d (X_{Sj}(t) - X_{ij}(t))^2}. \quad (7)$$

where $\|X_S(t) - X_i(t)\|_2$ represents the Euclidean distance between the dimensions of X_S and these of X_i . The mass of the Sun and object i at time t is simply calculated using the fitness evaluation as follows (considering a minimization problem):

$$M_s = r_2 \frac{fit_s(t) - worst(t)}{\sum_{k=1}^N (fit_k(t) - worst(t))}, \quad (8)$$

$$m_i = \frac{\text{fit}_i(t) - \text{worst}(t)}{\sum_{k=1}^N (\text{fit}_k(t) - \text{worst}(t))}, \quad (9)$$

where

$$fit_s(t) = best(t) = \min_{k \in 1, 2, \dots, N} fit_k(t), \quad (10)$$

$$worst(t) = \max_{k \in 1, 2, \dots, N} fit_k(t), \quad (11)$$

where r_2 is a number generated randomly between 0 and 1 to diverge the mass values for various planets. $\mu(t)$ is a function that exponentially decreases with time (t) to control search accuracy and is defined as follows:

$$\mu(t) = \mu_0 \times \exp(-\gamma \frac{t}{T_{max}}), \quad (12)$$

where γ is a constant; μ_0 is an initial value; and t and T_{max} are the current iteration number and maximum number of iterations, respectively.

Step 3: Calculating an object's velocity

The velocity of an object depends on its position relative to the Sun. That is, a planet's velocity increases if it is close to the Sun and decreases if it is far from it. If an object is close to the Sun, then the Sun's gravity is considerably strong, and the planet attempts to increase its speed to avoid being pulled toward the Sun. However, if an object is far from the Sun, then its velocity will slow down because the Sun's gravity is weak. Mathematically, this behavior is formulated in Eq. (13) to compute the velocity of an object around the Sun according to the vis-viva equation [103]. This equation is twofold. The first fold determines the velocities of planets close to the Sun by multiplying the distance between the current solution and a randomly selected solution, or the distance between two solutions that are randomly selected from the current population. This helps KOA diversify its search strategies. However, the diversity of the population's solutions during the optimization process may be minimized, and thus, velocity may be minimized in cases wherein a planet is close to the Sun. Accordingly, another step size based on the distance between the lower and upper bounds of the optimization problem is integrated into the first fold to assist in preserving the velocity of planets throughout the optimization process and avoiding being stuck in local minima. Second, on the basis of the proposition that planets are far from the Sun, the equation computes the velocity of the planet in accordance with the distance between a randomly selected solution and the current solution to reduce the velocity of planets compared with the first fold. The major shortcoming in the second fold is the lack of diversity between solutions, which may minimize the opportunity for KOA to escape local optima because changes in the current solution are too small. To address this flaw, a second step size based on the distance between the lower and upper bounds of the optimization issue is incorporated into the second fold.

$$V_i(t) = \begin{cases} \ell \times \left(2r_4 \vec{X}_i - \vec{X}_b \right) + \vec{e} \times \left(\vec{X}_a - \vec{X}_b \right) + (1 - R_{i-norm}(t)) \\ \quad \times \mathcal{F} \times \vec{U}_1 \times \vec{r}_5 \times \left(\vec{X}_{i,up} - \vec{X}_{i,low} \right), & \text{if } R_{i-norm}(t) \leq 0.5 \\ r_4 \times \mathcal{L} \times \left(\vec{X}_a - \vec{X}_i \right) + (1 - R_{i-norm}(t)) \\ \quad \times \mathcal{F} \times U_2 \times \vec{r}_5 \times \left(r_3 \vec{X}_{i,up} - \vec{X}_{i,low} \right), & \text{Else} \end{cases} \quad (13)$$

$$\ell = \vec{U} \times \mathcal{M} \times \mathcal{L}, \quad (14)$$

$$\mathcal{L} = \left[\mu(t) \times (M_S + m_i) \left| \frac{2}{R_i(t) + \varepsilon} - \frac{1}{a_{i(t)} + \varepsilon} \right| \right]^{\frac{1}{2}}, \quad (15)$$

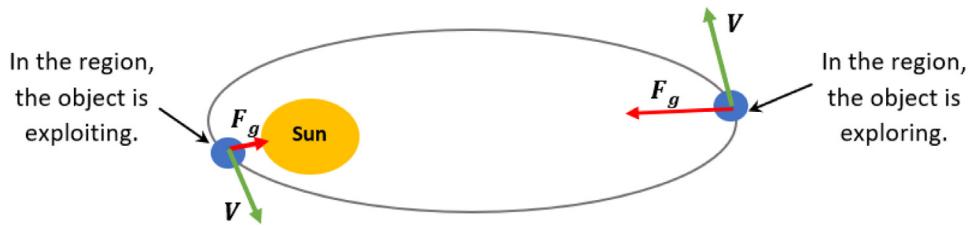


Fig. 4. Exploration and exploitation regions in the search space.

$$\mathcal{M} = (r_3 \times (1 - r_4) + r_4), \quad (16)$$

$$\vec{U} = \begin{cases} 0 & \vec{r}_5 \leq \vec{r}_6 \\ 1 & \text{Else,} \end{cases} \quad (17)$$

$$\mathcal{F} = \begin{cases} 1, & \text{if } r_4 \leq 0.5 \\ -1, & \text{Else,} \end{cases} \quad (18)$$

$$\vec{\epsilon} = (1 - \vec{U}) \times \vec{\mathcal{M}} \times \mathcal{L}, \quad (19)$$

$$\vec{\mathcal{M}} = (r_3 \times (1 - \vec{r}_5) + \vec{r}_5), \quad (20)$$

$$\vec{U}_1 = \begin{cases} 0 & \vec{r}_5 \leq r_4 \\ 1 & \text{Else,} \end{cases} \quad (21)$$

$$\vec{U}_2 = \begin{cases} 0 & r_3 \leq r_4 \\ 1 & \text{Else,} \end{cases} \quad (22)$$

where $\vec{V}_i(t)$ represents the velocity of object i at time t , \vec{X}_i represent object i , r_3 and r_4 are randomly generated numerical values at interval $[0, 1]$, and \vec{r}_5 and \vec{r}_6 are two vectors that include random values between 0 and 1. X_a and X_b represent solutions that are selected at random from the population; M_s and m_i represent the mass of X_s and X_i , respectively; $\mu(t)$ represents the universal gravitational constant; ϵ is a small value for preventing a divide-by-zero error; $R_i(t)$ represents the distance between the best solution X_s and the object X_i at time t ; and a_i represents the semimajor axis of the elliptical orbit of object i at time t , and it is defined by Kepler's third law mentioned in Eq. (2), as follows:

$$a_i(t) = r_3 \times \left[T_i^2 \times \frac{\mu(t) \times (M_s + m_i)}{4\pi^2} \right]^{\frac{1}{3}}, \quad (23)$$

where T_i represents the orbital period of object i and is determined by Eq. (5). In our proposed algorithm, the semimajor axis of the elliptical orbit of object i is assumed to decrease gradually with generations wherein the solutions move toward the promising region in which the global best solution is likely to be found. $R_{i-norm}(t)$ represents normalizing the Euclidian distance between X_s and X_i , and it is defined as follows:

$$R_{i-norm}(t) = \frac{R_i(t) - \min(R(t))}{\max(R(t)) - \min(R(t))}. \quad (24)$$

The purpose of Eq. (15) is to calculate the percentage of steps that each object will change. If $R_{i-norm}(t) \leq 0.5$, then the object is close to the Sun and will increase its speed to prevent drifting toward the Sun because of the latter's tremendous gravitational force. Otherwise, the object will slow down.

Step 4: Escaping from the local optimum

In the solar system, most objects revolve counterclockwise around the Sun, and they all rotate on their own axes; however, some objects revolve around the Sun in a clockwise direction. The proposed algorithm uses this behavior to escape from local

optimum regions. The proposed KOA simulates this behavior by using a flag \mathcal{F} that changes the search direction such that agents have a good chance of scanning the search space accurately.

Step 5: Updating objects' positions

As mentioned earlier, objects revolve around the Sun in their own elliptical orbits. During rotation, objects move closer to the Sun for a certain time and then move away from it. The proposed algorithm simulates this behavior through two major phases: the exploration and exploitation phases. KOA explores objects far from the Sun to find new solutions, while using solutions close to the Sun more accurately as it searches for new places near the best solutions. Fig. 4 shows the regions of exploration and exploitation around the Sun. The exploration and exploitation phases are subsequently described in detail. In the exploration phase, the objects are far from the Sun, indicating that the proposed algorithm explores the entire search area more efficiently. In accordance with the previous steps, a new position of each object far from the Sun is updated using Eq. (25):

$$\vec{X}_i(t+1) = \vec{X}_i(t) + \mathcal{F} \times \vec{V}_i(t) + (\mathbf{F}_{g_i}(t) + |\mathbf{r}|) \times \vec{U} \times (\vec{X}_s(t) - \vec{X}_i(t)), \quad (25)$$

where $\vec{X}_i(t+1)$ is the new position of object i at time $t+1$, $\vec{V}_i(t)$ is the velocity of object i required to reach the new position, $X_s(t)$ is the best position of the Sun found thus far, and \mathcal{F} is used as a flag to change search direction. Eq. (25) simulates the gravitational force of the Sun to the planets, where this equation employs another step size on the basis of calculating the distance between the Sun and the current planet multiplied by the gravitational force of the Sun to help KOA explore the regions around the best-so-far solution and find better outcomes in less number of function evaluations. In general, the velocity of planets will represent the exploration operator of KOA when a planet is far from the Sun. However, this velocity is affected by the gravitational force of the Sun, which helps the current planet slightly exploit regions near the optimal solution. Meanwhile, when a planet approaches the Sun, its velocity increases dramatically, allowing it to escape the Sun's gravitational pull. In such case, velocity represents local optimum avoidance if the best-so-far solution, referred to as the sun, is local minima, and the Sun's gravitational pull represents the exploitation operator to assist KOA in attacking the best-so-far solution to find better solutions.

Step 6: Updating distance with the Sun

To further improve the exploration and exploitation operators of planets, we attempt to mimic the typical behavior of the distance between the Sun and planets, which naturally varies over time. When planets are close to the Sun, KOA will focus on optimizing the exploitation operator; when the Sun is far, KOA will optimize the exploration operator. These rules depend on the value of the regulating parameter h , which varies gradually with the present time as shown in Fig. 5. When this value is large, the exploration operator is employed to expand planetary orbital separation from the Sun; conversely, when this value is small, the exploitation operator is used to exploit the regions

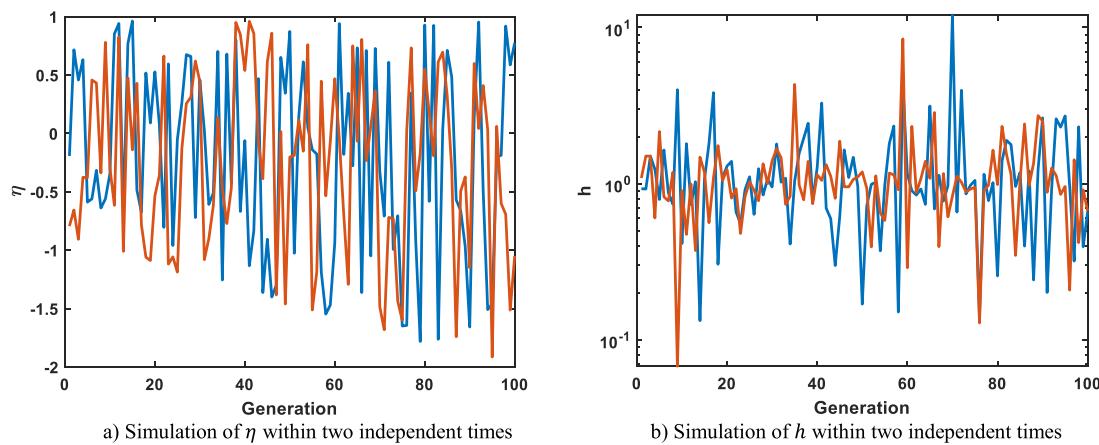
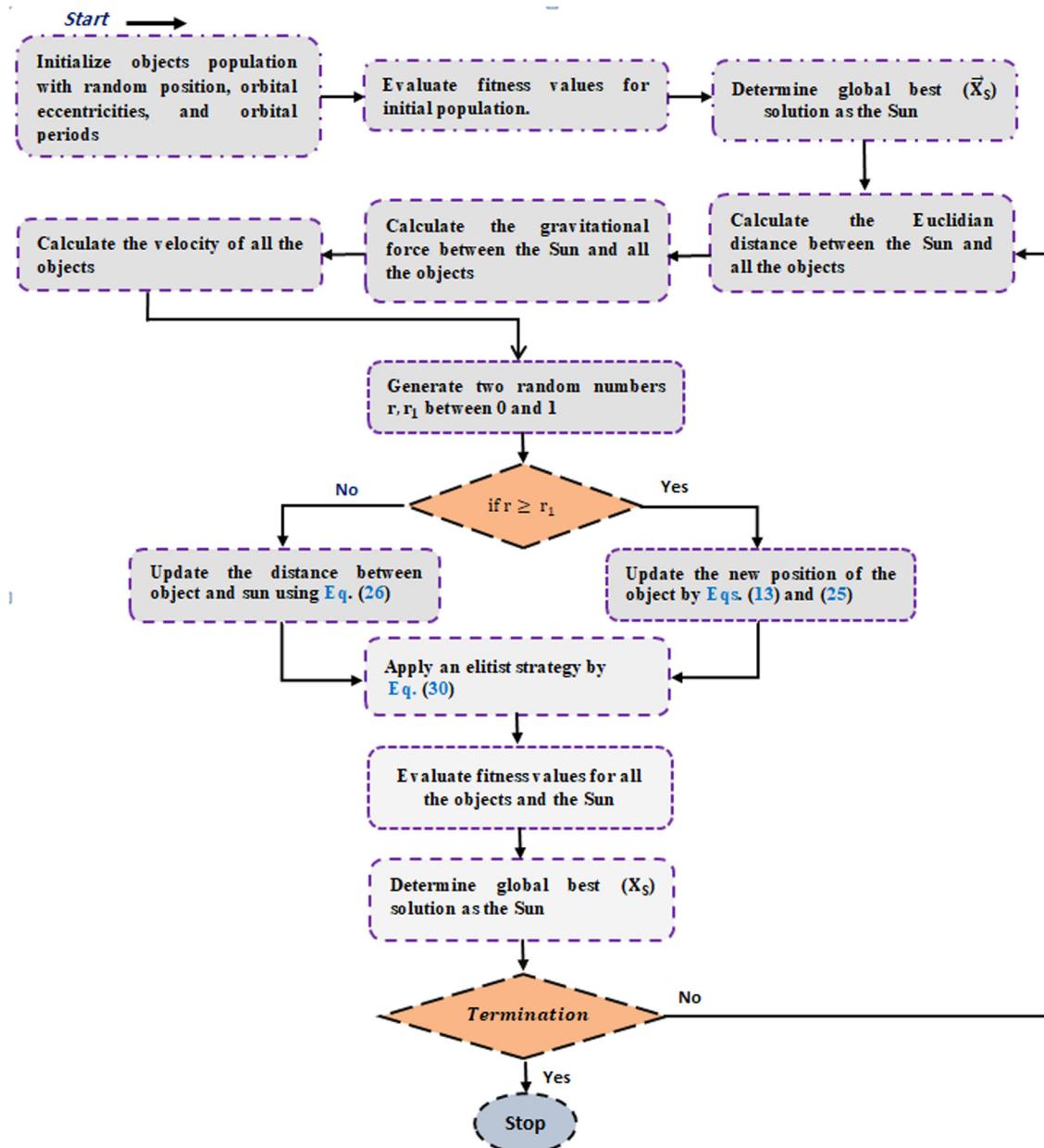
Fig. 5. Depiction the values of h based on the η factor within two independent times.

Fig. 6. Flowchart of KOA.

Table 2

Parameter settings of the rival optimizers and proposed KOA.

Algorithms	Parameters	Value	Algorithms	Parameters	Value
Well-studied, highly-cited optimizers					
GWO [29]	Convergence constant a N'	Decreases Linearly from 2 to 0 120	WOA [17]	Convergence constant a b N	Decreases Linearly from 2 to 0 1 120
Recently-published optimizers					
GTO [20]	p	0.03	POA [86]	R	0.2
	Beta	3		N	25
	w	8			
	N	25	DO [89]	α	[0, 1]
	k	[0, 1]		N	25
	N	25	COA [90]	N	25
FLA [92]	N	25	MGO [87]	N	25
	C_1, C_2, C_3, C_4, C_5	0.5, 2, 0.1, 0.2, 2			
	D	0.01	SO [88]	N	25
	T, T_1	0.25, 0.6		a	20
	C_1, C_2, C_3	0.5, 0.05, 2		b	12
SMA [34]	z	0.03		N	25
	N	25	KOA	N	25
				\bar{T}, μ_0, γ	3, 0.1, 15

around the best-so-far solution if the distance between the Sun and planets is small. This principle is randomly exchanged with Eq. (25) to improve the exploration and exploitation operators of KOA further, as listed in Algorithm 1. The mathematical model of this principle is described as follows:

$$\vec{X}_i(t+1) = \vec{X}_i(t) \times \vec{U}_1 + \left(1 - \vec{U}_1\right) \times \left(\frac{\vec{X}_i(t) + \vec{X}_S + \vec{X}_a(t)}{3.0} \right. \\ \left. + h \times \left(\frac{\vec{X}_i(t) + \vec{X}_S + \vec{X}_a(t)}{3.0} - \vec{X}_b(t) \right) \right), \quad (26)$$

where h is an adaptive factor for controlling the distance between the Sun and the current planet at time t , as defined below:

$$h = \frac{1}{e^{\eta r}}, \quad (27)$$

where r is a number that is generated randomly on the basis of the normal distribution, while η is a linearly decreasing factor from 1 to -2 , as defined below:

$$\eta = (a_2 - 1) \times r_4 + 1, \quad (28)$$

where a_2 is a cyclic controlling parameter that is decreasing gradually from -1 to -2 for \bar{T} cycles within the whole optimization process as defined below:

$$a_2 = -1 - 1 \times \left(\frac{t \% \frac{T_{max}}{\bar{T}}}{\frac{T_{max}}{\bar{T}}} \right). \quad (29)$$

Step 7: Elitism

This step implements an elitist strategy to ensure the best positions for planets and the Sun. This procedure is summarized using Eq. (30):

$$\vec{X}_{i,new}(t+1) = \begin{cases} \vec{X}_i(t+1), & \text{if } f(\vec{X}_i(t+1)) \leq f(\vec{X}_i(t)) \\ \vec{X}_i(t), & \text{Else} \end{cases}. \quad (30)$$

5. Results and discussion

This section evaluates the performance of the proposed optimizer by employing four recent challengeable mathematical test suites: CEC-2014, CEC-2017, CEC-2020, and CEC-2022 [108–110].

The CEC-2014 benchmark has a total of 30 test functions, while the CEC-2017 benchmark has a total of 29 test functions. The third benchmark has a total of 10 functions, while the fourth benchmark has a total of 23 test functions. These test functions can be broken down into four categories: unimodal, multimodal, hybrid, or composition. Their purpose is to present a significant obstacle for both the previously proposed and the newly proposed metaheuristic algorithms.

In addition, this study extensively compares the KOA against three different categories of cutting-edge optimizers to gauge its performance. The first category includes some of the most recently published algorithms like SO [88], FLA [92], COA [90], POA [86], DO [89], MGO [87], RUN, GTO, and SMA; the second category contains some of well-studied and highly-cited algorithms, such as WOA and GWO; the last category includes two high performance optimizers: LSHADE-cnEpSin [109], and LSHADE-SPACMA [111]. All of these competing algorithms were created to address the global optimization problem studied here; as a result, their controlling parameters were set at values recommended by the study's authors, with the exception of N and T_{max} , which were set at 25 and 200,000, respectively, to ensure a fair comparison. Tabulated in Table 2 are the stated parameter settings for each competing method.

The tests in this study were run on a computer equipped with MATLAB R2019a, Intel(R) Core(TM) i7-4700MQ processor running at 2.40 GHz, 32 GB of RAM, and a 64-bit version of Windows 10 Pro. Because all algorithms investigated in this study are stochastic, they are executed 30 independent times and the obtained fitness values within these runs are analyzed in terms of average, standard deviation (SD), and rank metric to determine the order of each algorithm. In addition, the Wilcoxon rank-sum test is used to show whether there is a difference between the KOA's outcomes and those of the rival optimizers. Finally, the convergence curve is employed to determine which method gets to the global solution the quickest.

5.1. Sensitivity analysis

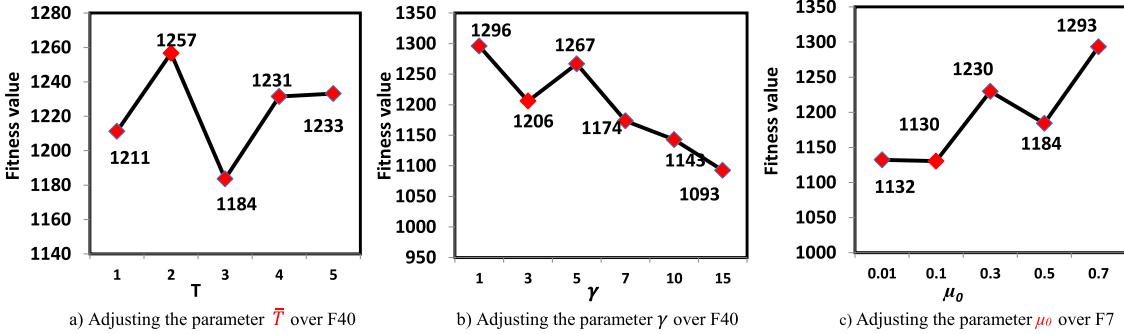
The KOA has three primary effective controlling parameters that need to be accurately calibrated in order to get the highest possible level of performance. These parameters are: \bar{T} , which defines the number of cycles; μ_0 , which determines the beginning value for the parameter μ ; and γ , which specifies the depth of the

Algorithm 1 Pseudo code of KOA.

```

Start
    Set parameters  $N, T_{max}, \mu_0, \gamma, \bar{T}$ .
    Initialize objects population with random position, orbital eccentricities, and orbital periods using Eq. (3), Eqs. (4), and (5), respectively.
    Evaluate fitness values for initial population.
    Determine the global best ( $X_S$ ) solution as the Sun.
While ( $t < T_{max}$ )
    Update  $e_i$   $i = 1, 2, \dots, N$ ,  $best(t)$ ,  $worst(t)$ , and  $\mu(t)$ , using Eqs. (10), (11) and (12), respectively.
    For  $i=1:N$ 
        Calculate the Euclidian distance between the Sun and the object  $i$  using Eq. (7).
        Calculate the gravitational force between the Sun and the object  $i$  using Eq. (6).
        Calculate the velocity of the object  $X_i$  using Eq. (13).
        Generate two random numbers  $r, r_1$  between 0 and 1.
        If  $r > r_1$  /* Update position of the planet*/
            Update the object position using Eq. (25).
        Else /*update the distance between the planet and the sun*/
            Update the object position using Eq. (26).
        End if
        Apply an elitist strategy to select the best position of the Sun and the objects, using Eq.(30)
         $t = t + 1$ 
    End for
End while
Stop

```

**Fig. 7.** KOA's parameter tuning.

linearly decreasing parameter μ . The more these parameters are accurately tuned, the better it could perform for the optimization problems. Therefore, Extensive trials under a broad range of values for each parameter have been conducted, and the findings of these studies are depicted in Fig. 7. This figure shows that the KOA's performance is maximized when \bar{T} , γ , and μ_0 are set to 3, 15, and 0.1, respectively.

5.2. Comparison on challengeable CEC2014 benchmark

In this section, we compare KOA's results to those of eleven optimizers using the CEC2014 test suite, which includes mathematical functions from four families [23]: unimodal (F1–F3), multimodal (F4–F16), composition (F17–F22), and hybrid (F23–F30). Because they contain numerous local optima, multimodal test functions are well-suited for validating the exploration operator of the stochastic optimizers. Unimodal test functions are recommended for evaluating the exploitation operator since they have just one global optimal solution, while composition and hybrid functions are utilized to evaluate local optima avoidance by an optimization algorithm. The number of dimensions for all test functions of this test suite is set to 10.

All algorithms are independently executed 30 times for each test function to return thirty fitness values, which are analyzed using three statistic information: Avg, SD, and rank metric to

reveal their performance within those independent runs, as reported in Tables 3 and 4. Inspecting these tables reveals that KOA is superior to all for all statistical information on 21 out of 30 test functions, and a little inferior to some for the remaining test functions. To show how far the obtained outcomes are close to the best-known value for each test function, an additional metric known as an overall effective percentage metric is used to compute the summation of error values between the best-known values and obtained values for all test functions. The mathematical equation of this metric is as follows:

$$\text{OE} = \sum_{i=1}^{T_f} |F_i^* - F_i| \quad (31)$$

where F_i^* indicates the best-known fitness value for the i th test functions, and F_i stands for the average fitness value obtained by an optimization algorithm for the same function. The value of this metric for each algorithm is shown in Fig. 8, which affirms that the proposed KOA is the nearest to the best-known value since it could reach the lowest normalized OE value, followed by MGO as the second best, while COA is the worst. Additionally, the summation of SD values reported in Tables 3 and 4 is shown in Fig. 9 to show the most stable algorithm. The lower the SD value is, the more stable the algorithm is. According to this figure, the stability of KOA is substantially competitive with some rival optimizers and significantly better than the others.

Table 3

Comparison over CEC2014 benchmark: F1–F16.

F1			F2			F3			F4			
Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	
KOA	1.00E+02	0.00E+00	1	2.00E+02	0.00E+00	1	3.00E+02	0.00E+00	1	419.27	16.92	2
SO	2.15E+07	1.50E+07	11	1.85E+09	1.00E+09	11	1.19E+04	3.46E+03	10	597.26	110.75	11
FLA	1.42E+05	8.43E+04	7	7.23E+03	4.60E+03	7	1.05E+03	6.95E+02	8	420.56	18.56	3
COA	1.48E+08	9.90E+07	12	6.74E+09	2.68E+09	12	1.44E+04	1.88E+03	11	2330.10	1213.95	12
MGO	1.47E+03	8.27E+02	3	2.29E+03	3.03E+03	3	3.17E+02	1.51E+01	3	424.78	15.58	6
GTO	1.06E+02	1.10E+01	2	2.00E+02	2.42E+01	2	3.00E+02	3.37E+07	2	429.42	12.23	9
RUN	1.90E+04	9.94E+03	4	4.43E+03	5.39E+03	4	4.78E+02	2.57E+02	6	418.54	15.45	1
GWO	4.30E+06	3.82E+06	10	1.11E+07	4.99E+07	9	3.94E+03	3.93E+03	9	433.42	11.12	10
WOA	2.70E+06	2.60E+06	9	8.14E+03	5.57E+03	8	2.71E+04	1.29E+04	12	428.12	19.42	8
SMA	8.33E+04	4.18E+04	6	4.85E+03	3.75E+03	6	3.60E+02	1.14E+02	5	427.15	14.10	7
DO	5.63E+04	4.54E+04	5	4.62E+03	5.25E+03	5	3.39E+02	7.31E+01	4	422.47	15.36	5
POA	3.62E+05	6.85E+05	8	4.00E+07	1.19E+08	10	8.48E+02	5.38E+02	7	422.27	16.06	4
F5			F6			F7			F8			
Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	
KOA	518.03	6.12	1	6.00E+02	1.96E-03	1	700.04	0.02	1	8.00E+02	0.00E+00	1
SO	520.45	0.24	12	6.09E+02	1.40E+00	11	732.72	21.34	11	8.49E+02	1.10E+01	11
FLA	520.01	0.01	3	6.03E+02	1.36E+00	3	700.19	0.06	3	8.00E+02	3.59E-04	2
COA	520.35	0.09	11	6.11E+02	7.65E-01	12	817.31	30.10	12	8.80E+02	1.07E+01	12
MGO	519.41	3.45	2	6.03E+02	1.67E+00	4	700.12	0.07	2	8.06E+02	2.59E+00	5
GTO	520.10	0.08	9	6.04E+02	1.69E+00	7	700.37	0.22	7	8.21E+02	8.25E+00	8
RUN	520.06	0.09	8	6.05E+02	1.45E+00	9	700.23	0.12	5	8.15E+02	6.36E+00	7
GWO	520.34	0.05	10	6.02E+02	1.27E+00	2	700.90	0.74	9	8.08E+02	5.07E+00	6
WOA	520.05	0.09	7	6.07E+02	1.67E+00	10	700.62	0.32	8	8.36E+02	1.18E+01	10
SMA	520.04	0.04	6	6.04E+02	1.62E+00	5	700.20	0.10	4	8.00E+02	1.82E+01	3
DO	520.01	0.03	5	6.04E+02	1.27E+00	6	700.32	0.21	6	8.02E+02	1.21E+00	4
POA	520.01	0.01	4	6.05E+02	1.39E+00	8	710.39	14.95	10	8.31E+02	1.15E+01	9
F9			F10			F11			F12			
Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	
KOA	904.39	1.89	1	1002.89	2.22	2	1873.96	202.77	7	1200.37	8.17E-02	8
SO	949.53	9.86	11	2146.39	286.69	11	2956.60	262.24	11	1201.55	2.72E-01	12
FLA	912.17	4.97	3	1000.31	0.60	1	1766.08	275.11	5	1201.13	5.54E-02	2
COA	977.51	15.30	12	2713.56	202.43	12	2966.09	223.58	12	1201.13	2.55E-01	11
MGO	912.34	4.89	4	1119.69	74.18	4	1727.95	270.77	4	1200.19	1.06E-01	5
GTO	928.30	13.13	7	1413.31	265.90	8	1810.76	232.36	6	1200.25	1.28E-01	6
RUN	935.72	6.81	9	1173.30	114.96	6	1557.80	226.97	2	1200.17	1.07E-01	4
GWO	912.56	5.71	5	1283.53	137.89	7	1548.32	316.12	1	1200.60	4.90E-01	10
WOA	944.66	19.01	10	1490.64	315.38	9	2575.27	420.20	10	1200.57	2.48E-01	9
SMA	911.01	4.11	2	1088.67	76.86	3	1966.42	277.43	8	1200.12	6.71E-02	1
DO	918.37	8.27	6	1146.62	97.74	5	2076.90	406.02	9	1200.26	1.36E-01	7
POA	930.71	8.03	8	1614.67	241.10	10	1667.68	222.91	3	1200.15	7.38E-02	3
F13			F14			F15			F16			
Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	
KOA	1.30E+03	2.91E-02	1	1.40E+03	2.82E-02	2	1500.90	0.21	1	1601.55	0.33	1
SO	1.30E+03	7.02E-01	11	1.41E+03	4.50E+00	11	1778.94	531.09	11	1603.46	0.23	12
FLA	1.30E+03	1.03E-01	6	1.40E+03	7.57E-02	5	1501.40	0.50	5	1602.35	0.36	4
COA	1.30E+03	1.02E+00	12	1.43E+03	9.86E+00	12	5706.75	5513.96	12	1603.43	0.14	11
MGO	1.30E+03	5.49E-02	3	1.40E+03	5.96E-02	3	1501.09	0.45	3	1602.17	0.43	2
GTO	1.30E+03	1.22E-01	7	1.40E+03	2.21E-01	8	1501.91	1.05	6	1603.00	0.40	9
RUN	1.30E+03	1.66E-01	9	1.40E+03	1.44E-01	9	1502.35	1.34	8	1602.59	0.47	6
GWO	1.30E+03	5.52E-02	2	1.40E+03	2.06E-01	6	1501.38	0.67	4	1602.38	0.55	5
WOA	1.30E+03	1.96E-01	10	1.40E+03	2.04E-01	7	1504.35	2.75	10	1603.23	0.36	10
SMA	1.30E+03	5.99E-02	4	1.40E+03	4.88E-02	1	1501.01	0.38	2	1602.32	0.45	3
DO	1.30E+03	8.20E-02	5	1.40E+03	7.33E-02	4	1502.20	1.08	7	1602.71	0.40	8
POA	1.30E+03	1.51E-01	8	1.40E+03	2.33E-01	10	1502.53	1.36	9	1602.69	0.41	7

Finally, Table 5 displays the Wilcoxon rank-sum test's p-values for comparing the results of each competing algorithm to KOA on each test function from the CEC2014. In light of the fact that the p-values in this table favor the null hypothesis that there are differences between the results, it may be concluded that KOA produces significantly different results from those achieved by competing optimizers for the vast majority of test functions. Since KOA outperformed numerous rival algorithms on the vast majority of the CEC2014 test functions, it deserves the label of highly-performing optimizer.

5.3. Comparison on challengeable CEC2017 benchmark

The CEC2017 challenging test suite is used here to further investigate the KOA's reliability. This benchmark, like CEC2014, consists of 29 mathematical test functions of varying difficulties to disclose the performance of optimization algorithms. These functions fall into four categories [23]: unimodal (F31–F32), multimodal (F33–F39), composition (F40–F49), and hybrid (F50–F59). Tables 6 and 7 detail the results of the Avg, SD, and rank metric statistics obtained from analyzing the thirty fitness values returned by each algorithm in thirty independent runs on this

Table 4

Comparison over CEC2014 benchmark: F17–F30.

F17			F18			F19			F20			
Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	
KOA	1.71E+03	5.61E+00	1	1800.46	0.64	1	1.90E+03	3.92E−02	1	2000.14	1.33E−01	1
SO	2.11E+05	2.08E+05	11	13899.89	14485.71	10	1.91E+03	1.02E+01	11	7961.86	4.52E+03	12
FLA	3.24E+04	6.80E+04	9	9031.87	5250.01	6	1.90E+03	4.13E−01	2	5382.99	3.65E+03	10
COA	3.54E+05	2.42E+05	12	28218.53	64543.55	12	1.93E+03	2.00E+01	12	6907.33	5.00E+03	11
MGO	4.38E+03	2.07E+03	5	8912.33	3797.46	4	1.90E+03	6.44E−01	3	2020.04	1.39E+01	2
GTO	2.23E+03	3.22E+02	2	1843.68	25.15	2	1.90E+03	1.07E+00	6	2028.30	2.48E+01	3
RUN	3.64E+03	1.09E+03	4	8993.48	3982.24	5	1.90E+03	7.62E−01	8	2049.20	2.89E+01	5
GWO	2.43E+04	1.02E+05	8	9709.31	4988.09	7	1.90E+03	5.66E−01	5	4594.90	3.56E+03	8
WOA	9.37E+04	1.26E+05	10	13053.70	9700.90	9	1.91E+03	1.48E+00	10	5240.92	2.65E+03	9
SMA	8.13E+03	4.94E+03	7	16471.40	12505.24	11	1.90E+03	5.34E−01	4	2121.31	4.33E+02	7
DO	5.45E+03	2.48E+03	6	10880.76	6692.76	8	1.90E+03	8.65E−01	7	2043.88	2.10E+01	4
POA	3.01E+03	1.85E+03	3	1924.96	60.28	3	1.90E+03	1.09E+00	9	2080.88	5.05E+01	6
F21			F22			F23			F24			
Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	
KOA	2100.21	0.18	1	2200.20	0.16	1	2569.04	65.69	2	2509.78	3.14	1
SO	1.91E+05	6.83E+05	12	2368.11	79.82	11	2656.58	33.53	7	2573.06	12.78	9
FLA	9754.37	7097.13	9	2219.55	37.65	3	2629.46	0.00	5	2526.97	7.20	3
COA	11140.57	12314.67	10	2419.49	104.74	12	2500.00	0.00	1	2600.00	0.00	12
MGO	2427.97	290.00	4	2222.76	28.62	4	2500.00	0.00	1	2569.62	36.01	7
GTO	2409.28	219.45	3	2232.50	31.10	5	2500.00	0.00	1	2578.88	28.81	10
RUN	4604.75	2045.05	7	2270.14	54.38	9	2500.00	0.00	1	2570.81	30.60	8
GWO	7864.22	4283.90	8	2268.70	51.96	8	2631.98	3.04	6	2533.32	28.72	5
WOA	14621.82	10963.65	11	2243.79	18.74	6	2616.89	39.64	3	2564.47	25.34	6
SMA	2332.52	340.53	2	2218.72	5.95	2	2500.00	0.00	1	2522.83	8.54	2
DO	3081.82	1235.64	6	2261.38	57.24	7	2625.14	23.64	4	2532.35	8.16	4
POA	2494.30	189.92	5	2280.48	56.47	10	2500.00	0.00	1	2582.14	28.04	11
F25			F26			F27			F28			
Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	
KOA	2616.74	8.27	1	2700.13	0.03	2	2701.78	0.51	1	3185.07	40.42	3
SO	2695.21	10.20	8	2700.82	0.42	10	2949.91	155.54	8	3189.58	235.23	4
FLA	2677.39	32.56	3	2700.24	0.08	8	3017.58	150.72	10	3058.65	120.45	2
COA	2700.00	0.00	12	2708.54	17.53	12	2900.00	0.00	7	3000.00	0.00	1
MGO	2697.96	11.17	11	2700.10	0.05	1	2847.21	89.04	3	3000.00	0.00	1
GTO	2695.74	12.49	10	2700.17	0.08	4	2809.21	98.74	2	3000.00	0.00	1
RUN	2695.67	11.49	9	2700.17	0.07	5	2848.31	87.18	4	3000.00	0.00	1
GWO	2684.25	28.24	4	2703.46	18.24	11	3005.43	125.31	9	3218.25	57.99	5
WOA	2688.46	14.39	5	2700.43	0.23	9	3090.64	138.59	12	3315.20	128.23	7
SMA	2675.16	33.79	2	2700.17	0.04	6	2867.08	74.88	6	3000.00	0.00	1
DO	2690.93	20.58	6	2700.22	0.12	7	3068.58	126.41	11	3256.42	65.68	6
POA	2692.64	16.66	7	2700.15	0.08	3	2861.04	79.26	5	3000.00	0.00	1
F29			F30									
Avg	SD	Rank	Avg	SD	Rank							
KOA	3094.22	38.85	1	3496.11	41.09	2						
SO	58309.31	1.42E+05	8	5591.63	1311.59	12						
FLA	61001.67	3.15E+05	9	3788.08	287.42	5						
COA	3100.02	0.10	2	3200.00	0.01	1						
MGO	3288.43	44.06	4	3736.51	400.29	4						
GTO	3212.41	106.07	3	3903.35	561.28	6						
RUN	3539.63	440.49	6	4088.73	538.82	9						
GWO	2.72E+05	6.99E+05	12	4012.26	493.27	7						
WOA	1.19E+05	4.37E+05	10	4905.04	724.95	11						
SMA	3503.23	379.10	5	3605.82	180.86	3						
DO	3579.76	548.07	7	4062.87	277.88	8						
POA	2.19E+05	8.70E+05	11	4146.49	450.87	10						

Bold font indicates the best findings.

benchmark. According to the data presented in these tables, KOA performs better than all for 25 of the 29 test functions, and slightly worse than some for the other four functions. Fig. 10 presents the value of the OE metric for each algorithm, demonstrating how near the achieved results are to the best-known value for each test function. Examination of this figure verifies that the proposed KOA is the most close to the best-known value, as it could achieve the lowest normalized OE value, followed by MGO as the second best. Moreover, the most stable approach is depicted in Fig. 11 by adding the SD values from Tables 6 and 7. This figure indicates that KOA's stability is superior to that of any competing algorithm. Finally, Table 8 presents the p-values

returned from the Wilcoxon rank-sum test that compares the results of each competing method to KOA on each test function in this benchmark. The p-values in this table suggest that there are significant differences between the results produced by KOA and those attained by rival optimizers for the vast majority of test functions. Similar to its performance on CEC2014, KOA achieved better results than all competing algorithms on the great majority of the CEC2017 test functions; so, it is fair to call it a top-performing optimizer.

Table 5

Wilcoxon rank-sum test for KOA against other rival optimizers using CEC2014 test suite.

Fun	SO	FLA	COA	MGO	GTO	RUN	GWO	WOA	SMA	DO	POA
F1	1.397E-11										
F2	1.212E-12										
F3	1.212E-12										
F4	1.930E-11	7.040E-03	1.930E-11	3.741E-04	7.008E-06	3.864E-03	8.631E-09	7.845E-05	2.133E-06	1.633E-04	2.048E-03
F5	3.812E-10	9.503E-06	3.012E-11	5.825E-03	4.376E-01	2.890E-03	3.012E-11	9.464E-03	2.890E-03	1.782E-04	1.166E-05
F6	5.215E-12										
F7	3.020E-11	3.020E-11	3.020E-11	2.602E-08	4.077E-11	9.919E-11	7.389E-11	3.020E-11	6.696E-11	3.020E-11	3.020E-11
F8	1.212E-12	1.212E-12	1.212E-12	1.175E-12	1.210E-12	1.212E-12	1.212E-12	1.212E-12	1.212E-12	1.212E-12	1.212E-12
F9	3.020E-11	8.891E-10	3.020E-11	3.808E-09	3.685E-11	3.020E-11	1.174E-09	3.020E-11	4.183E-09	6.066E-11	3.020E-11
F10	3.020E-11	3.497E-09	3.020E-11	3.012E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11	1.010E-08	3.690E-11	3.020E-11
F11	2.112E-11	1.321E-10	1.220E-8	1.022E-4	3.020E-11	2.521E-8	3.020E-11	3.020E-11	2.112E-05	3.020E-11	3.020E-11
F12	3.020E-11	6.066E-11	3.020E-11	9.063E-08	2.839E-04	2.602E-08	3.790E-01	8.120E-04	8.153E-11	3.182E-04	3.820E-10
F13	3.020E-11	8.891E-10	3.020E-11	3.988E-04	1.411E-09	3.690E-11	8.771E-02	4.975E-11	1.892E-04	2.377E-07	1.558E-08
F14	3.020E-11	1.850E-08	3.020E-11	7.659E-05	9.063E-08	4.975E-11	3.564E-04	2.602E-08	6.627E-01	1.868E-05	4.998E-09
F15	3.020E-11	6.283E-06	3.020E-11	7.483E-02	1.249E-05	1.010E-08	1.221E-02	4.200E-10	4.553E-01	5.462E-09	6.121E-10
F16	3.020E-11	2.034E-09	3.020E-11	7.599E-07	1.206E-10	9.756E-10	1.473E-07	3.020E-11	3.646E-08	8.153E-11	2.154E-10
F17	3.020E-11	3.020E-11	3.020E-11	3.020E-11	4.975E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11
F18	3.020E-11										
F19	3.020E-11	3.020E-11	3.020E-11	3.338E-11	3.020E-11						
F20	3.020E-11										
F21	3.020E-11										
F22	3.020E-11	7.389E-11	3.020E-11	8.993E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11	3.690E-11	3.020E-11	3.020E-11
F23	9.039E-11	1.620E-11	6.113E-07	6.113E-07	6.113E-07	6.113E-07	6.113E-07	1.620E-11	7.615E-09	6.113E-07	8.198E-11
F24	3.020E-11	3.020E-11	1.212E-12	3.020E-11	1.618E-11	3.020E-11	2.610E-10	3.020E-11	1.464E-10	3.338E-11	1.444E-11
F25	2.519E-11	6.718E-10	1.211E-12	1.924E-12	5.215E-12	4.108E-12	3.472E-10	2.980E-11	2.183E-10	3.687E-11	2.780E-11
F26	3.020E-11	5.533E-08	3.020E-11	2.608E-02	6.097E-03	4.676E-02	8.534E-01	2.922E-09	2.959E-05	6.765E-05	4.464E-01
F27	3.020E-11	5.494E-11	1.212E-12	7.040E-08	1.956E-11	9.400E-11	1.613E-10	3.020E-11	2.960E-10	4.200E-10	7.879E-12
F28	1.484E-01	2.088E-05	1.189E-12	1.189E-12	1.189E-12	1.189E-12	2.704E-02	2.356E-07	1.189E-12	1.847E-06	1.689E-12
F29	3.020E-11	3.020E-11	1.601E-01	3.020E-11	7.389E-11	1.152E-07	3.018E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11
F30	3.020E-11	7.380E-10	1.720E-12	1.892E-04	2.439E-09	1.521E-05	6.722E-10	3.020E-11	1.174E-03	4.975E-11	7.380E-10

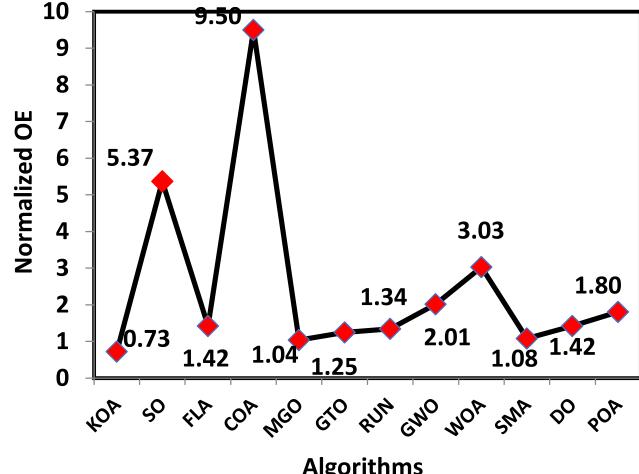
Bold font are the p-values smaller than 5%.

Fig. 8. OE on all CEC2014 test functions.

5.4. Comparison on challengeable CEC2020 benchmark

In this section, the CEC-2020 test suite is used to conduct additional evaluations of the proposed algorithm and its competitors. This suite is made up of ten different test functions, all of which fall into one of four categories [23]: unimodal, multimodal, composition, or hybrid. The capacity of an algorithm to explore, exploit, and avoid local minima can be evaluated with the help of these test functions. The characteristics of this test suite are found in elsewhere. Table 9 displays the average, rank, and standard deviation values obtained from 30 separate runs of each algorithm on each test function. These results demonstrate that KOA's performance is greater for eight out of the ten CEC-2020

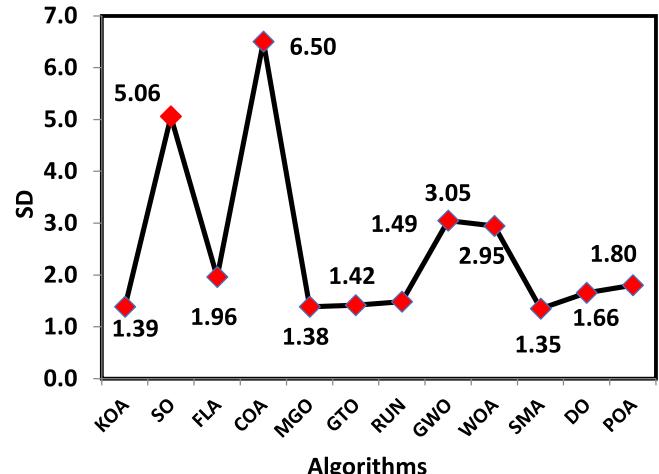


Fig. 9. SD on all CEC2014 test functions.

test functions. In addition, the findings of the Wilcoxon rank-sum tests are presented in Table 10 to disclose the difference between the outcomes of KOA and those of other optimizers. The p-values in this table show that the alternative hypothesis is true for most of the test functions. This means that the proposed KOA method produces results that are very different from those produced by a competing method. This section offers more experimental evidence to support the contention that KOA belongs in the class of potent optimizers.

5.5. Comparison on challengeable CEC2022 benchmark

On the CEC2022 test suite, both the proposed method and other approaches undergo additional testing. This suite consists of 12 different test functions, each of which can be classified as

Table 6

Comparison over CEC2017 benchmark: F31–F46.

F31			F32			F33			F34			
Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	
KOA	1.00E+02	0.00E+00	1	3.00E+02	0.00E+00	1	4.00E+02	0.00E+00	1	5.04E+02	1.55E+00	1
SO	2.53E+09	2.03E+09	11	8.19E+03	2.39E+03	10	6.06E+02	1.34E+02	11	5.53E+02	1.26E+01	11
FLA	8.38E+03	4.38E+03	7	3.03E+02	2.97E+00	6	4.15E+02	2.55E+01	8	5.11E+02	3.76E+00	2
COA	1.27E+10	4.35E+09	12	1.12E+04	2.62E+03	11	1.30E+03	3.93E+02	12	5.95E+02	2.52E+01	12
MGO	2.25E+03	2.22E+03	3	3.00E+02	4.22E−14	1	4.00E+02	1.23E−01	3	5.12E+02	4.75E+00	3
GTO	8.57E+02	9.50E+02	2	3.00E+02	3.32E−12	2	4.00E+02	3.59E−02	2	5.26E+02	1.04E+01	7
RUN	2.92E+03	1.09E+03	4	3.00E+02	2.10E−04	5	4.00E+02	3.04E−01	4	5.29E+02	1.05E+01	8
GWO	2.40E+07	8.78E+07	9	1.29E+03	1.44E+03	9	4.17E+02	1.75E+01	9	5.16E+02	8.54E+00	5
WOA	5.77E+04	1.78E+05	8	3.86E+02	8.88E+01	8	4.24E+02	3.41E+01	10	5.53E+02	1.98E+01	10
SMA	7.39E+03	4.43E+03	6	3.00E+02	6.83E−05	4	4.03E+02	1.10E+00	6	5.13E+02	5.21E+00	4
DO	4.40E+03	2.90E+03	5	3.00E+02	4.75E−05	3	4.01E+02	6.95E−01	5	5.23E+02	6.60E+00	6
POA	1.06E+08	3.27E+08	10	3.60E+02	5.64E+01	7	4.07E+02	1.09E+01	7	5.46E+02	1.79E+01	9
F35			F36			F37			F38			
Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	
KOA	6.00E+02	0	1	7.15E+02	1.87E+00	1	8.04E+02	1.44E+00	1	9.00E+02	0.00E+00	1
SO	6.32E+02	1.16E+01	11	7.98E+02	1.44E+01	11	8.47E+02	7.57E+00	11	1.39E+03	2.19E+02	11
FLA	6.00E+02	1.53E−02	3	7.21E+02	4.00E+00	3	8.11E+02	4.74E+00	2	9.00E+02	9.96E−02	5
COA	6.51E+02	1.21E+01	12	8.23E+02	1.84E+01	12	8.62E+02	8.77E+00	12	1.72E+03	2.69E+02	12
MGO	6.00E+02	2.26E−01	4	7.27E+02	5.85E+00	4	8.15E+02	5.63E+00	5	9.00E+02	1.63E−02	3
GTO	6.05E+02	3.83E+00	7	7.47E+02	1.49E+01	7	8.24E+02	8.68E+00	7	9.73E+02	7.38E+01	7
RUN	6.13E+02	8.19E+00	8	7.54E+02	1.25E+01	8	8.25E+02	3.89E+00	9	1.02E+03	7.35E+01	8
GWO	6.00E+02	1.05E+00	5	7.28E+02	9.18E+00	5	8.15E+02	6.02E+00	4	9.11E+02	1.78E+01	6
WOA	6.30E+02	1.41E+01	10	7.71E+02	2.31E+01	10	8.39E+02	1.47E+01	10	1.23E+03	3.15E+02	10
SMA	6.00E+02	1.12E−02	2	7.19E+02	4.75E+00	2	8.12E+02	4.45E+00	3	9.00E+02	7.43E−05	2
DO	6.03E+02	3.71E+00	6	7.40E+02	9.25E+00	6	8.23E+02	8.50E+00	6	9.00E+02	8.42E−02	4
POA	6.17E+02	1.14E+01	9	7.66E+02	1.61E+01	9	8.25E+02	6.03E+00	8	1.13E+03	1.46E+02	9
F39			F40			F41			F42			
Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	
KOA	1.18E+03	1.24E+02	1	1.10E+03	4.28E−01	1	1.24E+03	5.13E+01	1	1.30E+03	2.50E+00	1
SO	2.52E+03	2.65E+02	11	2.80E+03	1.35E+03	11	1.94E+07	2.41E+07	11	3.60E+04	3.19E+04	11
FLA	1.41E+03	1.88E+02	2	1.13E+03	8.63E+01	7	3.83E+04	4.65E+04	8	1.13E+04	1.07E+04	8
COA	2.59E+03	1.71E+02	12	5.66E+03	5.93E+03	12	4.71E+08	4.20E+08	12	1.32E+06	3.40E+06	12
MGO	1.52E+03	2.52E+02	4	1.11E+03	7.56E+00	3	1.73E+04	1.34E+04	4	8.76E+03	5.72E+03	7
GTO	1.87E+03	3.10E+02	9	1.12E+03	1.12E+01	5	3.61E+03	3.33E+03	2	1.44E+03	1.31E+02	2
RUN	1.53E+03	1.54E+02	5	1.12E+03	7.52E+00	6	7.39E+03	4.75E+03	3	8.48E+03	2.91E+03	6
GWO	1.55E+03	2.07E+02	6	1.28E+03	8.01E+02	10	4.21E+05	6.15E+05	9	7.84E+03	4.40E+03	5
WOA	2.00E+03	3.24E+02	10	1.18E+03	9.78E+01	9	2.87E+06	3.42E+06	10	1.50E+04	1.09E+04	10
SMA	1.50E+03	2.03E+02	3	1.11E+03	4.43E+00	2	3.63E+04	2.25E+04	7	1.17E+04	1.20E+04	9
DO	1.62E+03	2.29E+02	7	1.12E+03	9.31E+00	4	2.69E+04	2.42E+04	6	6.23E+03	4.26E+03	4
POA	1.70E+03	3.22E+02	8	1.16E+03	4.10E+01	8	2.59E+04	3.46E+04	5	1.90E+03	3.18E+02	3
F43			F44			F45			F46			
Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	
KOA	1.40E+03	4.05E−01	1	1.50E+03	2.78E−01	1	1.60E+03	2.35E−01	1	1.70E+03	8.54E−01	1
SO	2.93E+03	2.27E+03	11	7.72E+03	3.78E+03	12	1.89E+03	1.21E+02	11	1.83E+03	6.13E+01	11
FLA	6.55E+03	6.83E+03	12	2.66E+03	1.17E+03	9	1.69E+03	9.93E+01	5	1.73E+03	3.97E+01	4
COA	1.54E+03	4.57E+01	9	6.83E+03	3.65E+03	11	2.15E+03	1.42E+02	12	1.83E+03	4.90E+01	12
MGO	1.42E+03	1.41E+01	2	1.51E+03	1.31E+01	3	1.66E+03	8.82E+01	2	1.73E+03	1.90E+01	2
GTO	1.45E+03	2.46E+01	5	1.53E+03	3.71E+01	5	1.69E+03	8.80E+01	4	1.75E+03	2.10E+01	6
RUN	1.49E+03	1.25E+02	7	1.53E+03	7.33E+00	4	1.75E+03	1.13E+02	8	1.75E+03	1.62E+01	8
GWO	2.14E+03	1.41E+03	10	2.75E+03	1.38E+03	10	1.72E+03	1.19E+02	7	1.74E+03	1.58E+01	5
WOA	1.51E+03	3.19E+01	8	2.52E+03	1.03E+03	8	1.81E+03	7.97E+01	10	1.79E+03	5.12E+01	10
SMA	1.43E+03	9.41E+00	3	1.51E+03	6.27E+00	2	1.69E+03	7.96E+01	3	1.73E+03	1.07E+01	3
DO	1.45E+03	1.73E+01	6	1.54E+03	1.62E+01	6	1.71E+03	1.19E+02	6	1.75E+03	2.62E+01	9
POA	1.45E+03	1.82E+01	4	1.59E+03	7.25E+01	7	1.78E+03	1.21E+02	9	1.75E+03	1.41E+01	7

unimodal, multimodal, hybrid, or composition [112]. The number of dimensions for the test functions of this benchmark is also set to 10. Table 11 presents the Avg, rank, and SD for each of the 30 runs that were performed independently by the proposed and rival optimizers on each test function. This table discloses that KOA performs better for 10 out of the 12 test functions. The Wilcoxon rank-sum test is performed to determine whether or not there is a significant difference between the results obtained by KOA and those obtained by each competing optimizer. According to the p-values presented in Table 12, there is a statistically significant difference between the results of KOA and those of the competing algorithms for the majority of the test functions. In this section,

it is demonstrated beyond a reasonable doubt that KOA falls into the category of extremely effective optimizers.

5.6. Comparison with some high-performing optimizers

This section presents a comparison between the performance of our proposed KOA and that of various highly-performing optimizers, like LSHADE-SPACMA and LSHADE-cnEpSin. The tests here used a population size of 40 individual for the proposed KOA and a total of 1,000,000 function evaluations for KOA, LSHADE-SPACMA, and LSHADE-cnEpSin. All other KOA parameters are left at their default values, whereas those used by the LSHADE-SPACMA and LSHADE-cnEpSin are as recommended in [109,111].

Table 7

Comparison over CEC2017 benchmark: F47–F59.

F47			F48			F49			F50									
Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank							
KOA	1.80E+03	1.69E−01	1	1.90E+03	1.17E−02	1	2.00E+03	5.70E−02	1	2.21E+03	2.73E+01	1						
SO	2.67E+05	6.33E+05	11	9.07E+03	5.62E+03	11	2.14E+03	6.13E+01	11	2.32E+03	5.44E+01	11						
FLA	1.24E+04	9.18E+03	5	7.25E+03	5.79E+03	10	2.00E+03	3.81E+00	2	2.28E+03	5.55E+01	6						
COA	3.45E+07	8.81E+07	12	7.10E+03	7.82E+03	9	2.25E+03	6.67E+01	12	2.39E+03	4.07E+01	12						
MGO	8.18E+03	6.03E+03	4	1.91E+03	6.01E+00	3	2.04E+03	4.54E+01	5	2.25E+03	5.74E+01	4						
GTO	1.90E+03	5.47E+01	2	1.94E+03	4.43E+01	6	2.06E+03	4.15E+01	6	2.21E+03	3.06E+01	2						
RUN	1.37E+04	9.93E+03	6	5.50E+03	5.00E+03	8	2.10E+03	5.33E+01	9	2.22E+03	4.07E+01	3						
GWO	2.40E+04	1.39E+04	10	4.79E+03	4.91E+03	7	2.08E+03	5.42E+01	7	2.31E+03	5.68E+00	10						
WOA	1.74E+04	1.11E+04	7	1.16E+04	1.03E+04	12	2.14E+03	6.02E+01	10	2.30E+03	6.63E+01	8						
SMA	1.85E+04	9.99E+03	8	1.91E+03	4.97E+00	2	2.02E+03	7.86E+00	3	2.29E+03	5.13E+01	7						
DO	1.87E+04	1.25E+04	9	1.92E+03	1.26E+01	4	2.03E+03	1.55E+01	4	2.31E+03	4.93E+01	9						
POA	2.32E+03	8.90E+02	3	1.93E+03	2.21E+01	5	2.10E+03	4.12E+01	8	2.27E+03	6.43E+01	5						
F51			F52			F53			F54									
Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank							
KOA	2.29E+03	2.94E+01	1	2.60E+03	2.66E+00	1	2659.004	111.5871	2	2924.44	23.17393	3						
SO	2.50E+03	1.10E+02	11	2.67E+03	1.46E+01	11	2777.891	40.79871	11	3068.016	76.41074	11						
FLA	2.30E+03	1.11E+01	4	2.62E+03	6.55E+00	4	2761.012	11.19381	8	2934.814	27.79822	7						
COA	3.37E+03	5.90E+02	12	2.73E+03	3.48E+01	12	2919.992	58.53015	12	3615.941	268.8972	12						
MGO	2.30E+03	1.36E+01	2	2.61E+03	5.62E+00	2	2734.61	44.84893	5	2934.909	20.23169	8						
GTO	2.30E+03	1.74E+01	3	2.63E+03	1.64E+01	7	2643.963	128.1244	1	2933.987	24.12454	6						
RUN	2.30E+03	1.45E+01	5	2.62E+03	7.58E+00	6	2705.285	93.5871	4	2925.817	23.31772	4						
GWO	2.30E+03	2.23E+01	6	2.62E+03	6.03E+00	3	2743.664	11.89724	6	2925.902	16.66221	5						
WOA	2.40E+03	3.35E+02	10	2.64E+03	6.62E+01	9	2763.537	53.66107	10	2943.692	26.27209	10						
SMA	2.40E+03	2.88E+02	9	2.62E+03	5.49E+00	5	2755.849	8.621534	7	2937.606	25.61746	9						
DO	2.35E+03	2.14E+02	8	2.63E+03	1.09E+01	8	2761.976	51.25171	9	2918.36	23.30972	2						
POA	2.34E+03	5.52E+01	7	2.65E+03	2.40E+01	10	2662.081	139.2828	3	2917.468	60.17531	1						
F55			F56			F57			F58									
Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank							
KOA	2890	54.77226	1	3092.776	2.611114	3	3117.865	101.3035	1	3148.146	7.518682	2						
SO	3468.613	268.2337	11	3136.682	27.59024	11	3423.95	116.4292	11	3317.968	59.90061	10						
FLA	3048.463	306.781	5	3092.073	3.548945	2	3368.953	127.425	8	3179.939	38.98517	3						
COA	4406.307	368.596	12	3201.24	47.70254	12	3785.404	94.09487	12	3440.719	118.4767	12						
MGO	2968.565	87.5256	3	3093.942	3.116389	5	3176.753	47.83581	3	3193.675	41.95896	6						
GTO	2957.482	90.66191	2	3095.041	3.094395	6	3259.594	149.5934	7	3194.698	39.53903	7						
RUN	2997.625	177.032	4	3093.552	1.962371	4	3254.412	145.7105	5	3189.233	36.99788	4						
GWO	3117.283	368.4897	8	3103.138	22.45468	9	3413.657	76.97087	10	3190.798	45.82499	5						
WOA	3431.707	512.7826	10	3121.368	32.15657	10	3407.155	158.8932	9	3354.085	96.90034	11						
SMA	3161.601	415.2671	9	3090.161	1.524928	1	3171.926	116.1128	2	3146.665	13.15428	1						
DO	3080.995	376.902	6	3102.775	12.43903	8	3210.211	140.2073	4	3225.176	51.82298	9						
POA	3116.833	414.1887	7	3102.088	15.19062	7	3259.407	131.1139	6	3203.595	37.17605	8						
F59																		
Avg	SD	Rank																
KOA	3531.176	118.0533	1															
SO	2749404	3857822	11															
FLA	227407.5	346641.8	7															
COA	11167209	9550214	12															
MGO	114544.5	281738.2	4															
GTO	1143173	2260600	10															
RUN	9082.849	1579.99	3															
GWO	557997.8	620950.4	9															
WOA	274753.6	376334.9	8															
SMA	6051.013	1368.043	2															
DO	164845.1	353403.9	5															
POA	217145.4	428132.2	6															

Table 13 displays the Avg and SD that results from analyzing the fitness values obtained by independently executing each algorithm 30 times on the CEC2017 test functions. This table show that KOA outperforms LSHADE-SPACMA and LSHADE-cnEpSin on 15 out of the 30 test functions, while failing on six of them and being competitive on the other 8 test functions. Our findings are validated, suggesting that KOA is indeed a powerful optimizer.

5.7. Convergence speed analysis

This section shows the convergence curves of the proposed KOA against the other 11 optimizers on four families of test functions (see Fig. 12): unimodal, multimodal, hybrid, and composition. These test functions are based on a number of dimensions equal to 10. From Fig. 12, it can be seen that KOA's performance is significantly superior to all the rival algorithms for the unimodal test functions: F1, F2, and F3, because it could reach the lowest fitness value much faster than all the competitors. But, GTO, at the end of the optimization process, could compete the proposed algorithm in terms of the solution quality. On the multimodal test functions (F4, F9, F41, and F42) that are more difficult

Table 8

Wilcoxon rank-sum test for KOA against other rival optimizers using CEC2017 test suite.

Fun	SO	FLA	COA	MGO	GTO	RUN	GWO	WOA	SMA	DO	POA
F31	1.720E-12										
F32	1.720E-12	1.720E-12	1.720E-12	2.119E-05	7.787E-12	1.720E-12	1.720E-12	1.720E-12	1.720E-12	1.720E-12	1.720E-12
F33	1.212E-12										
F34	3.020E-11	5.573E-10	3.020E-11	1.111E-09	3.009E-11	3.020E-11	3.497E-09	3.020E-11	1.957E-10	3.020E-11	3.020E-11
F35	3.151E-12	3.151E-12	3.151E-12	1.320E-09	3.151E-12						
F36	3.020E-11	4.311E-08	3.020E-11	1.174E-09	3.020E-11	3.020E-11	6.518E-09	3.020E-11	7.739E-06	3.020E-11	3.020E-11
F37	3.020E-11	1.010E-08	3.020E-11	4.479E-11	3.016E-11	3.020E-11	6.696E-11	3.020E-11	8.891E-10	3.020E-11	3.020E-11
F28	1.720E-12	4.104E-11	1.720E-12	1.054E-04	1.926E-12	1.720E-12	2.697E-12	1.720E-12	4.562E-11	4.104E-11	1.720E-12
F39	3.020E-11	9.792E-05	3.020E-11	5.090E-06	2.154E-10	7.119E-09	7.773E-09	3.020E-11	9.063E-08	1.102E-08	2.015E-08
F40	2.020E-11										
F41	3.020E-11										
F42	3.020E-11										
F43	1.073E-11										
F44	3.020E-11	3.690E-11	3.020E-11	3.020E-11							
F45	3.020E-11	6.526E-07	3.020E-11	6.010E-08	9.919E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11
F46	3.020E-11	3.324E-06	3.020E-11	4.975E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11	8.993E-11	3.020E-11	3.020E-11
F47	3.020E-11										
F48	3.016E-11										
F49	1.720E-12	3.016E-12	1.720E-12	1.718E-12	1.720E-12						
F50	1.760E-10	4.626E-10	2.155E-11	3.739E-07	3.908E-07	2.004E-07	2.644E-11	4.626E-10	3.151E-10	1.312E-10	3.588E-09
F51	3.009E-11	4.186E-10	3.009E-11	2.383E-08	4.170E-09	5.554E-10	4.559E-10	5.588E-07	5.519E-08	4.983E-09	
F52	3.020E-11	5.573E-10	3.020E-11	6.722E-10	6.691E-11	3.690E-11	1.287E-09	5.573E-10	8.993E-11	3.020E-11	3.020E-11
F53	1.682E-09	2.993E-11	2.993E-11	5.259E-10	1.977E-02	4.091E-07	6.475E-09	1.165E-09	2.993E-11	2.353E-10	2.148E-02
F54	2.980E-11	1.852E-03	2.980E-11	4.687E-04	4.442E-04	1.324E-02	7.958E-01	1.020E-06	2.520E-04	1.259E-01	6.203E-01
F55	4.099E-12	4.099E-12	4.099E-12	3.622E-07	2.906E-08	7.936E-04	2.640E-06	2.248E-08	4.099E-12	2.684E-07	1.433E-04
F56	2.939E-11	9.315E-02	2.938E-11	1.955E-01	3.163E-03	1.666E-01	2.316E-02	1.147E-09	4.307E-05	1.700E-07	2.962E-07
F57	1.410E-10	1.044E-10	7.846E-12	3.625E-07	1.123E-08	2.550E-09	3.406E-11	1.401E-10	5.777E-08	1.714E-08	4.866E-09
F58	3.020E-11	1.585E-04	3.020E-11	8.485E-09	1.596E-07	2.831E-08	1.729E-06	3.020E-11	1.260E-01	1.957E-10	8.101E-10
F59	3.020E-11	3.020E-11	3.020E-11	3.338E-11	1.429E-08	3.020E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11	4.502E-11

Bold font are the p-values smaller than 5%.

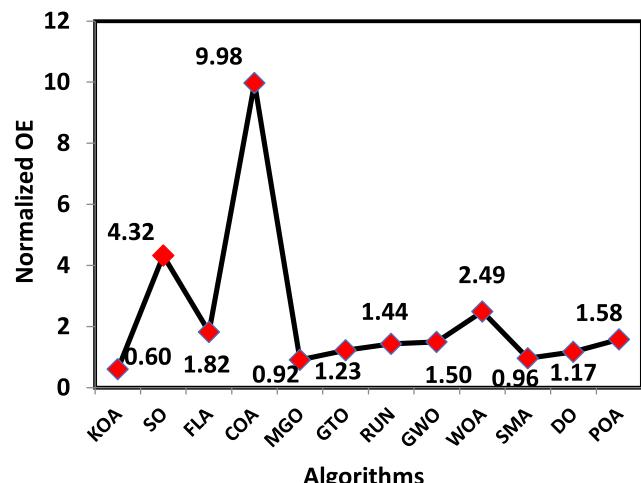


Fig. 10. OE on all CEC2017 test functions.

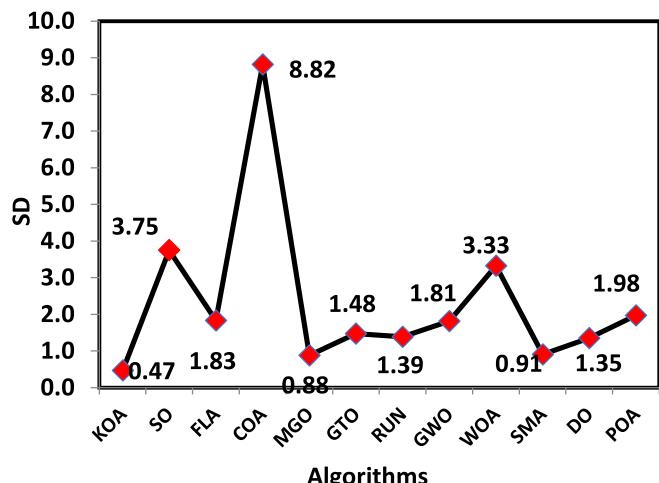


Fig. 11. SD on all CEC2017 test functions.

than the unimodal functions, KOA could be clearly differentiated compared to all the rival algorithms that could not compete with the proposed KOA even after completing the optimization process. On the hybrid and composition test function (F21, F25, and F27), the convergence accuracy of KOA rapidly diverges from the other algorithms. This shows that KOA is powerful not only for convergence speed but also for the final quality.

6. Application I: Constrained engineering problems

In this section, the effectiveness of KOA is evaluated in comparison to nine other optimizers on eight different constrained engineering problems under a population size and a number of function evaluations of 25 and 50,000, respectively. These problems include the welded beam design, tension/compression

spring design, pressure vessel design, 3-bar truss design, 10-bar truss design, tabular column design, cantilever beam design, and gear train design. It is not simple to find solutions to these constrained problems since they have a number of constraints that need to be satisfied before the optimization process can be considered complete. There are many different types of penalties that can be implemented in order to deal with these constraints [113]. In this study, one of the most common penalty methods, dubbed death penalty, is utilized to give infeasible solutions an objective value that is high enough for them to be eliminated throughout the optimization process if minimization is the goal.

The search boundary approach is a critical setting that needs to be fine-tuned so that the proposed KOA functions as efficiently as possible. There are two common search boundary methods for making solutions that are outside the search space feasible. The

Table 9
Comparison over CEC2020 benchmark.

F60			F61			F62			F63			
Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	
KOA	1.00E+02	0.00E+00	1	1.15E+03	4.97E+01	1	7.14E+02	2.08E+00	1	1900.52	0.07	6
SO	2.46E+09	1.44E+09	11	2.44E+03	2.48E+02	11	8.03E+02	1.97E+01	11	1900.26	0.57	4
FLA	9.12E+03	5.34E+03	7	1.38E+03	1.54E+02	2	7.19E+02	6.12E+00	2	1900.24	0.14	3
COA	1.10E+10	3.97E+09	12	2.56E+03	1.27E+02	12	8.32E+02	1.70E+01	12	1900.00	0.00	1
MGO	2.33E+03	2.77E+03	3	1.69E+03	1.94E+02	7	7.24E+02	5.02E+00	4	1900.00	0.00	1
GTO	1.43E+03	1.77E+03	2	1.87E+03	2.60E+02	9	7.50E+02	1.74E+01	7	1900.00	0.00	1
RUN	3.77E+03	1.80E+03	4	1.53E+03	1.82E+02	5	7.56E+02	1.41E+01	8	1900.00	0.00	1
GWO	4.61E+07	1.37E+08	9	1.52E+03	2.46E+02	4	7.28E+02	9.05E+00	5	1900.12	0.25	2
WOA	6.34E+04	2.27E+05	8	2.00E+03	3.02E+02	10	7.74E+02	2.13E+01	10	1900.00	0.00	1
SMA	8.83E+03	3.92E+03	6	1.46E+03	1.77E+02	3	7.22E+02	5.48E+00	3	1900.00	0.00	1
DO	5.18E+03	3.18E+03	5	1.68E+03	2.57E+02	6	7.42E+02	9.29E+00	6	1900.27	0.44	5
POA	1.41E+08	3.98E+08	10	1.73E+03	2.82E+02	8	7.70E+02	2.12E+01	9	1900.00	0.00	1
F64			F65			F66			F67			
Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	
KOA	1708.4397	7.1694	1	1.60E+03	3.58E-01	1	2100.10	0.09	1	2.30E+03	1.52E+01	1
SO	134432.70	177065.99	11	1.86E+03	8.73E+01	11	26369.93	32990.56	12	2.53E+03	2.19E+02	11
FLA	13685.95	25570.73	8	1.74E+03	1.01E+02	8	8688.41	6230.68	9	2.33E+03	1.45E+02	5
COA	375890.07	243483.08	12	2.06E+03	1.90E+02	12	10553.16	8555.98	10	3.27E+03	4.15E+02	12
MGO	4447.25	1874.14	5	1.70E+03	6.59E+01	2	2470.66	292.00	4	2.30E+03	1.69E+01	2
GTO	2163.06	242.91	2	1.73E+03	9.90E+01	6	2441.06	205.88	3	2.30E+03	1.35E+01	3
RUN	3802.09	1349.41	4	1.72E+03	6.27E+01	5	4291.96	1971.21	7	2.30E+03	1.28E+01	4
GWO	54084.24	124793.43	9	1.78E+03	1.08E+02	10	7679.92	4201.85	8	2.33E+03	8.42E+01	6
WOA	102861.92	169900.93	10	1.77E+03	9.40E+01	9	16988.60	11111.27	11	2.38E+03	2.71E+02	9
SMA	8989.66	4811.03	7	1.70E+03	5.04E+01	3	2292.43	336.23	2	2.38E+03	2.38E+02	8
DO	4765.63	2419.30	6	1.71E+03	6.94E+01	4	3144.53	1086.92	6	2.41E+03	2.94E+02	10
POA	2535.94	593.18	3	1.74E+03	6.71E+01	7	2482.08	255.77	5	2.33E+03	4.29E+01	7
F68			F69									
Avg	SD	Rank	Avg	SD	Rank							
KOA	2619.46	115.14	1	2919.62	22.84	3						
SO	2774.01	69.15	11	3035.06	42.58	11						
FLA	2750.39	48.86	7	2925.66	30.58	5						
COA	2916.21	69.23	12	3665.56	298.81	12						
MGO	2726.56	61.99	4	2937.55	20.21	10						
GTO	2670.74	123.10	2	2928.20	25.83	6						
RUN	2740.77	46.09	5	2924.36	23.81	4						
GWO	2743.03	13.29	6	2929.30	17.25	7						
WOA	2768.01	53.18	10	2935.23	26.28	9						
SMA	2752.36	7.62	8	2932.47	23.13	8						
DO	2763.04	51.30	9	2918.96	64.17	2						
POA	2672.64	138.70	3	2917.17	55.36	1						

Table 10
Wilcoxon rank-sum test for KOA against other rival optimizers using CEC2017 test suite.

Fun	SO	FLA	COA	MGO	GTO	RUN	GWO	WOA	SMA	DO	POA
F60	1.720E-12										
F61	3.020E-11	2.891E-03	3.020E-11	8.983E-11	8.153E-11	4.311E-08	3.831E-05	3.020E-11	5.859E-06	3.081E-08	2.034E-09
F62	3.020E-11	1.493E-04	3.020E-11	2.154E-10	3.020E-11	3.020E-11	2.034E-09	3.020E-11	5.092E-08	3.020E-11	3.020E-11
F63	4.238E-05	1.461E-10	1.212E-12	1.212E-12	1.212E-12	1.212E-12	3.886E-08	1.212E-12	1.212E-12	9.908E-05	1.212E-12
F64	3.020E-11	3.020E-11	3.020E-11	1.777E-10	3.020E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11	3.338E-11
F65	3.020E-11	3.474E-10	3.020E-11	3.338E-11	3.020E-11	3.020E-11	3.338E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11
F66	3.020E-11										
F67	2.916E-11	5.404E-10	2.916E-11	3.558E-08	4.918E-10	4.918E-10	8.262E-09	6.930E-09	5.404E-10	7.568E-09	5.404E-10
F68	3.203E-08	1.378E-10	2.829E-11	1.727E-09	4.765E-05	1.518E-10	2.532E-09	1.378E-10	2.829E-11	1.378E-10	1.376E-04
F69	2.978E-11	3.260E-02	2.978E-11	2.744E-06	2.372E-03	1.761E-03	2.704E-02	2.583E-05	2.520E-04	4.441E-04	1.833E-02

Bold font stand for the p-values smaller than 5%.

first method involves assigning the lower bound to the dimensions that are smaller and the upper bound to the dimensions that are larger; the second method requires creating new random values within the search bounds for the dimensions that are outside the search space of the problem. Following numerous tests, we determined that the randomly tradeoff between those two methods could highly maximize the performance of KOA for the constrained engineering problems.

6.1. The welded beam design problem

In this section, the welded beam design (WBD) problem shown in Fig. 13(a) is used to assess KOA's efficacy. Optimal values for four design variables (h , l , t , and b) must be accurately estimated in order to optimize/minimize the total fabrication cost of a welded beam while subjecting to seven constraints (bending stress, tip detection, weld coverage, buckling load, and cost). The outline of the formal formulation of the WBD issue in mathematics is presented below:

Table 11
Comparison over CEC2020 benchmark.

F70			F2			F3			F4			
Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	
KOA	3.00E+02	1.06E-14	1	4.02E+02	3.39E+00	1	6.00E+02	6.33E-14	1	804.44	1.62	1
SO	6.97E+03	2.63E+03	10	5.40E+02	6.21E+01	11	6.35E+02	9.26E+00	11	845.70	8.80	11
FLA	3.03E+02	2.42E+00	6	4.12E+02	1.91E+01	6	6.00E+02	1.07E-02	3	828.73	12.67	9
COA	7.46E+03	1.52E+03	11	1.94E+03	7.84E+02	12	6.56E+02	1.08E+01	12	864.02	8.05	12
MGO	3.00E+02	5.97E-14	1	4.07E+02	1.26E+01	4	6.00E+02	1.65E-02	2	812.50	3.96	3
GTO	3.00E+02	9.92E-13	2	4.08E+02	1.23E+01	5	6.05E+02	3.39E+00	7	825.11	8.92	7
RUN	3.00E+02	1.01E-03	5	4.04E+02	4.44E+00	2	6.12E+02	5.32E+00	8	822.55	5.11	5
GWO	1.74E+03	1.69E+03	8	4.17E+02	1.42E+01	9	6.01E+02	1.23E+00	5	812.38	4.59	2
WOA	5.15E+03	2.81E+03	9	4.17E+02	2.60E+01	10	6.29E+02	1.21E+01	10	839.70	17.37	10
SMA	3.00E+02	4.03E-05	4	4.06E+02	2.22E+00	3	6.00E+02	1.71E-02	4	822.59	10.22	6
DO	3.00E+02	4.59E-05	3	4.16E+02	2.51E+01	8	6.03E+02	3.19E+00	6	828.66	11.15	8
POA	3.49E+02	3.47E+01	7	4.14E+02	2.57E+01	7	6.19E+02	1.02E+01	9	819.76	4.90	4
F5			F6			F7			F8			
Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	
KOA	900.00	0.02	1	1.80E+03	3.83E-01	1	2000.02	0.11	1	2.20E+03	6.20E+00	1
SO	1283.41	134.49	10	6.24E+06	2.26E+07	11	2072.92	14.47	11	2.24E+03	2.30E+01	11
FLA	921.48	28.35	6	3.41E+03	1.86E+03	6	2015.73	8.07	2	2.22E+03	7.62E+00	2
COA	1671.24	230.75	12	2.18E+07	2.86E+07	12	2101.59	23.31	12	2.24E+03	1.36E+01	12
MGO	900.26	0.29	2	2.19E+03	5.40E+02	3	2019.93	7.47	4	2.22E+03	6.28E+00	3
GTO	935.19	53.15	7	1.84E+03	6.82E+01	2	2025.02	7.97	6	2.22E+03	6.19E+00	4
RUN	972.67	43.65	8	3.01E+03	1.12E+03	5	2038.89	10.80	9	2.22E+03	1.51E+00	9
GWO	908.02	12.66	4	5.70E+03	2.54E+03	9	2027.77	10.06	7	2.22E+03	6.81E+00	7
WOA	1314.84	392.66	11	3.51E+03	1.66E+03	7	2051.88	17.74	10	2.23E+03	5.27E+00	10
SMA	900.48	0.75	3	5.96E+03	1.62E+03	10	2017.15	7.60	3	2.22E+03	6.13E-01	5
DO	918.77	83.89	5	4.89E+03	1.91E+03	8	2022.35	3.87	5	2.22E+03	6.14E+00	8
POA	1100.37	148.11	9	2.36E+03	9.02E+02	4	2033.88	13.09	8	2.22E+03	7.77E+00	6
F9			F10			F11			F12			
Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	Avg	SD	Rank	
KOA	2529.28	0.00E+00	1	2500.21	0.04	1	2770.00	151.20	6	2864.32	1.16	5
SO	2627.59	4.05E+01	10	2561.51	82.23	9	3994.93	378.41	11	2897.50	22.29	11
FLA	2534.18	2.68E+01	7	2510.79	74.64	3	2774.37	184.86	7	2862.49	1.95	2
COA	2753.75	4.71E+01	11	2814.21	231.63	12	4314.66	336.35	12	2959.53	52.96	12
MGO	2529.28	0.00E+00	1	2526.84	48.88	5	2768.46	131.60	5	2864.54	1.82	6
GTO	2529.41	5.80E-01	4	2520.37	45.08	4	2650.11	91.10	1	2863.98	1.37	4
RUN	2529.28	1.52E-04	3	2534.96	53.42	6	2750.86	153.45	3	2863.91	1.05	3
GWO	2561.80	3.60E+01	9	2565.87	55.98	10	2919.95	136.87	10	2868.81	13.86	9
WOA	2533.54	1.36E+01	5	2542.29	64.97	7	2869.35	86.88	9	2876.21	17.15	10
SMA	2529.28	4.44E-05	2	2504.02	20.47	2	2755.02	175.36	4	2861.20	1.60	1
DO	2534.18	2.68E+01	6	2566.67	60.07	11	2806.68	150.71	8	2865.35	2.12	7
POA	2536.83	2.76E+01	8	2550.43	62.05	8	2749.02	194.46	2	2865.60	2.85	8

Table 12
Wilcoxon rank-sum test for KOA against other rival optimizers using CEC2022 test suite.

Fun	SO	FLA	COA	MGO	GTO	RUN	GWO	WOA	SMA	DO	POA
F1	1.720E-12	1.720E-12	1.720E-12	2.875E-09	2.140E-12	1.720E-12	1.720E-12	1.720E-12	1.720E-12	1.720E-12	1.720E-12
F2	1.860E-11	8.468E-08	1.860E-11	3.815E-05	1.184E-06	7.741E-05	4.419E-10	4.581E-07	1.385E-07	2.253E-06	3.162E-05
F3	1.015E-11	1.015E-11	1.015E-11	1.350E-04	1.015E-11						
F4	3.018E-11	3.687E-11	3.018E-11	1.188E-10	1.087E-10	3.018E-11	4.180E-09	3.018E-11	3.687E-11	3.018E-11	3.018E-11
F5	2.366E-12	2.366E-12	2.366E-12	7.752E-12	2.366E-12	2.366E-12	2.366E-12	2.366E-12	7.949E-12	5.728E-12	2.366E-12
F6	3.020E-11										
F7	1.929E-11	2.366E-11	1.929E-11	1.929E-11	1.929E-11	1.929E-11	1.929E-11	1.929E-11	2.137E-11	1.929E-11	1.929E-11
F8	3.020E-11	2.377E-07	3.020E-11	1.385E-06	5.092E-08	7.389E-11	2.872E-10	5.494E-11	1.850E-08	8.101E-10	6.722E-10
F9	1.212E-12	1.212E-12	1.212E-12	Nan	8.152E-02	1.212E-12	1.212E-12	1.212E-12	1.212E-12	1.212E-12	1.212E-12
F10	3.020E-11	7.659E-05	3.020E-11	1.174E-09	1.094E-10	8.153E-11	1.957E-10	3.020E-11	1.873E-07	8.993E-11	1.206E-10
F11	2.229E-11	8.246E-03	2.229E-11	2.207E-01	5.354E-01	3.793E-03	5.793E-08	5.793E-08	1.690E-02	3.183E-05	5.368E-01
F12	2.734E-11	2.915E-04	2.734E-11	5.884E-02	6.293E-02	1.146E-01	1.251E-01	2.227E-08	8.965E-10	2.207E-02	3.348E-02

Bold font stand for the p-values smaller than 5%.

- Solution representation:

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

- Objective function:

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

- Formulation of the constraints:

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

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

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

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

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

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

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

Table 13

Comparison with some highly-performing optimizers using CEC2017.

F31		F32		F33		F34		
Avg	SD	Avg	SD	Avg	SD	Avg	SD	
KOA	100	0	300	0	400	0	500.5306	0.5049
LSHADE-SPACMA	100	0	300	0	400	0	501.1930	0.6293
LSHADE-cnEpSin	100	0	300	0	400	0	501.2930	0.6715
	F35		F36		F37		F38	
Avg	SD	Avg	SD	Avg	SD	Avg	SD	
KOA	600	0	711.6367	1.3066	800.5638	0.6229	900	0
LSHADE-SPACMA	600	0	710.9970	0.3639	800.7950	0.6293	900	0
LSHADE-cnEpSin	600	0	711.4840	0.2770	801.7940	0.7814	900	0
	F39		F40		F41		F42	
Avg	SD	Avg	SD	Avg	SD	Avg	SD	
KOA	1006.978	7.5057	1100	0	1200.0656	0.1012	1302.3886	2.5754
LSHADE-SPACMA	1007.620	6.4316	1100	0	1321.6000	77.1133	1302.9700	2.5650
LSHADE-cnEpSin	1017.320	37.0358	1100	0	1301.6573	54.0510	1303.3860	2.3366
	F43		F44		F45		F46	
Avg	SD	Avg	SD	Avg	SD	Avg	SD	
KOA	1400	0	1500.008	0.0209	1600.133	0.1533	1700.0147	0.0569
LSHADE-SPACMA	1400.2900	0.6715	1500.3500	0.2068	1600.6600	0.2963	1700.0500	0.0982
LSHADE-cnEpSin	1400	0	1500.2041	0.1749	1600.5801	0.4735	1700.0782	0.1010
	F47		F48		F49		F50	
Avg	SD	Avg	SD	Avg	SD	Avg	SD	
KOA	1800.0502	0.0990	1900	0	2000	0	2200	0
LSHADE-SPACMA	1804.3000	8.4059	1900.0700	0.0509	2006.2800	9.6110	2200.0000	3.03E-13
LSHADE-cnEpSin	1802.3409	6.3829	1900.1520	0.4737	2000.0937	0.1508	2251.4720	54.2618
	F51		F52		F53		F54	
Avg	SD	Avg	SD	Avg	SD	Avg	SD	
KOA	2287.9288	31.875	2580.8894	76.3693	2496.6667	18.2574	2925.4902	22.7880
LSHADE-SPACMA	2300.0300	0.0906	2601.1300	1.4595	2706.9300	72.7150	2925.2200	23.4039
LSHADE-cnEpSin	2300.0000	2.14E-13	2600.5458	1.1529	2683.0861	96.5093	2925.6101	23.4325
	F55		F56		F57		F58	
Avg	SD	Avg	SD	Avg	SD	Avg	SD	
KOA	2900.0000	0.0000	3092.1026	2.2237	3103.8576	21.1289	3133.7738	3.6787
LSHADE-SPACMA	2900.0000	0.0000	3089.5200	0.0000	3100.0000	0.0000	3127.5400	0.8631
LSHADE-cnEpSin	2900.0000	0.0000	3089.0908	1.5958	3116.9786	35.7940	3127.3157	1.0206
	F59							
Avg	SD							
KOA	3435.6943	41.5279						
LSHADE-SPACMA	3425.0300	33.1098						
LSHADE-cnEpSin	3415.5161	32.4314						

Bold values stands for the best findings.

Where

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

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

The results achieved by KOA and those of 11 optimizers are listed in Table 14. These results were acquired by running each optimizer 30 independent times on the problem. This table illustrates that KOA is superior to all optimizers in terms of Avg, SD, and Rank, and that it is competitive with some competitors for the best fitness value. The p-value is displayed in the same table as another observation to illustrate the contrast between the results obtained by KOA and those obtained by the competing optimizers. According to the p-value displayed in this table, there is a statistically significant difference between the findings of KOA and those of each competing optimizer. An illustration is shown in Fig. 13(b) to disclose the convergence speed of the proposed KOA against the rival optimizers on the WBD problem. This figure demonstrates that KOA converges faster.

6.2. Tension/compression spring design problem

A tension/compression spring is depicted in Fig. 14(a), and the objective of this challenge is to lessen the weight of the

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

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

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

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

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

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

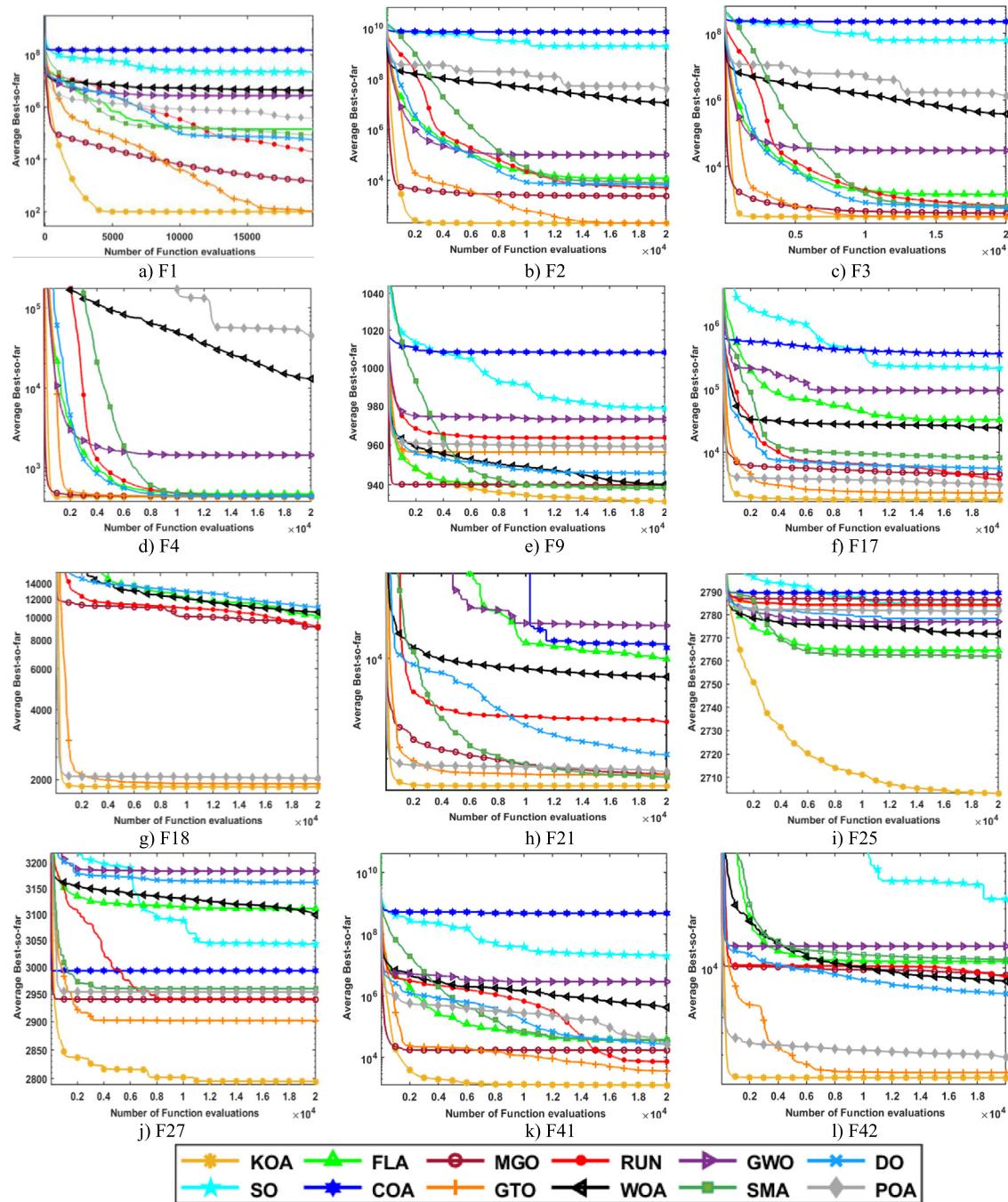


Fig. 12. The convergence curves of KOA and the other 11 rival optimizers on some test functions.

Table 14

Comparison using Welded beam design problem.

Algorithms	X ₁	X ₂	X ₃	X ₃	Best	Avg	SD	Rank	p-value
KOA	0.205720	3.470704	9.036624	0.205730	1.7248658492	1.724866	1.82E-15	1	7.868E-08
FLA	0.191322	3.909485	9.027901	0.206269	1.7625873489	2.831815	6.54E-01	9	1.887E-11
COA	0.174041	7.087014	8.997138	0.207648	2.1324620263	3.080523	5.44E-01	10	6.556E-11
GTO	0.205720	3.470704	9.036624	0.205730	1.7248658492	1.776438	1.31E-01	7	1.887E-11
RUN	0.205720	3.470699	9.036623	0.205730	1.7248692196	1.763343	3.40E-02	6	1.887E-11
GWO	0.205570	3.474292	9.036827	0.205758	1.7253668597	1.726182	8.32E-04	4	1.887E-11
WOA	0.188567	3.847065	9.117049	0.206380	1.7666804561	2.206351	5.26E-01	8	1.887E-11
SMA	0.205699	3.471142	9.036628	0.205730	1.7248973587	1.725941	1.98E-03	3	1.887E-11
DO	0.205723	3.470664	9.036533	0.205734	1.7248818489	1.726894	4.16E-03	5	1.887E-11
POA	0.205720	3.470704	9.036624	0.205730	1.7248658492	1.724893	1.30E-04	2	1.887E-11

Bold values stands for the best findings.

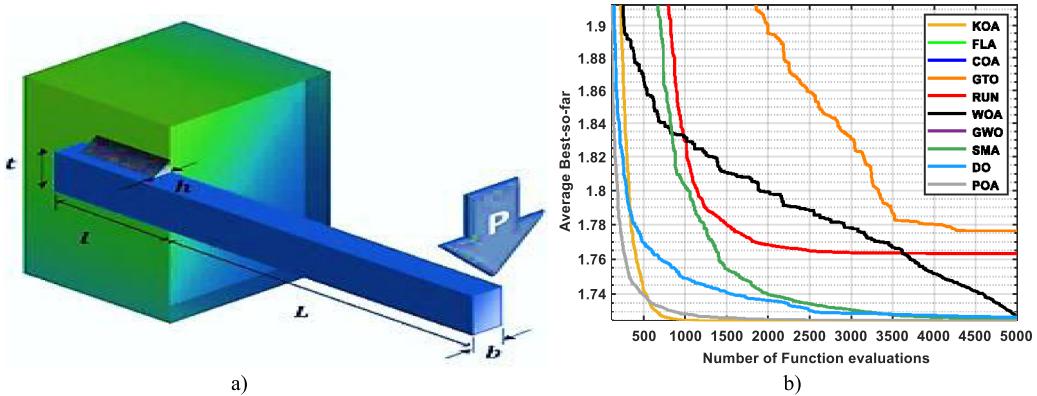


Fig. 13. The welded beam design problem: (a) Schematic [30] and (b) KOA's Convergence curve.

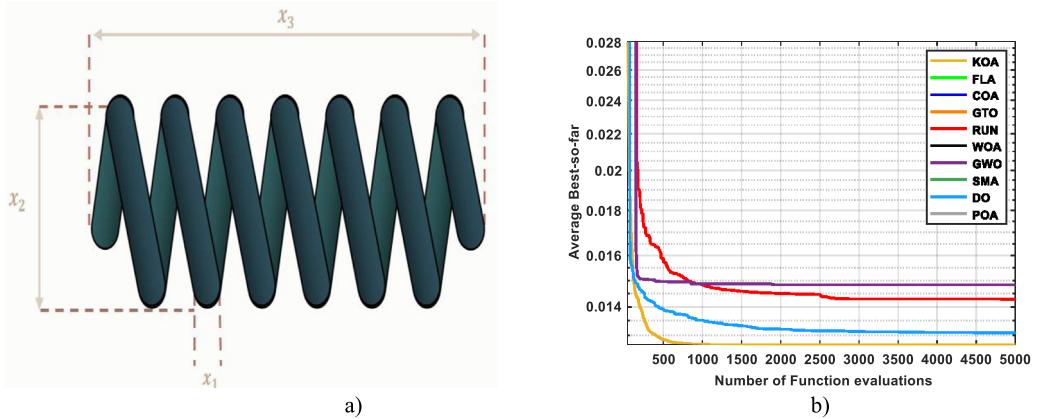


Fig. 14. Tension/compression spring design: (a) Schematic; (b) KOA's Convergence curve.

spring [113,114]. The optimal design must be capable of satisfying a number of constraints, including shear stress, surge frequency, and deflection. The mean coil diameter (D), the wire diameter (d), and the number of active coils (N) are all design variables that need to be correctly assessed in order to produce the best possible design. The mathematical model for this problem is written down as follows:

- Solution representation:

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

- Objective function:

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

- Formulation of constraints:

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

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

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

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

$$\text{Variable range } 0.05 \leq x_1 \leq 2.00$$

$$0.25 \leq x_2 \leq 1.30$$

$$2.00 \leq x_3 \leq 15.0$$

Table 15 contains the near-optimal fitness value and weight of three design variables, in addition to some other statistical

information such as Avg, Rank, p -value, and SD that were acquired by KOA and 11 competing methods. The data presented in this table reveals that KOA is the most effective strategy because it has a higher Best, Avg, SD, and Rank value. In addition, the convergence rate of the proposed algorithm compared to all the rival optimizers on this problem is displayed in Fig. 14(b). This figure shows that KOA is faster.

6.3. Pressure vessel design problem

The objective of this issue is to reduce total output by determining the optimal values for four design variables, taking into account pressure requirements as an optimization constraint. Design variables include the inner radius (R), the thickness of the shell (T_s), the thickness of the head (Th), and the length of the cylindrical component (L). Fig. 15(a) depicts the structure of the pressure vessel. This problem's mathematical model is as follows [34]:

- Solution representation:

Consider

$$X = [x_1 \ x_2 \ x_3 \ x_4] = [T_s \ Th \ R \ L]$$

- Objective function:

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

- Formulation of constraints:

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

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

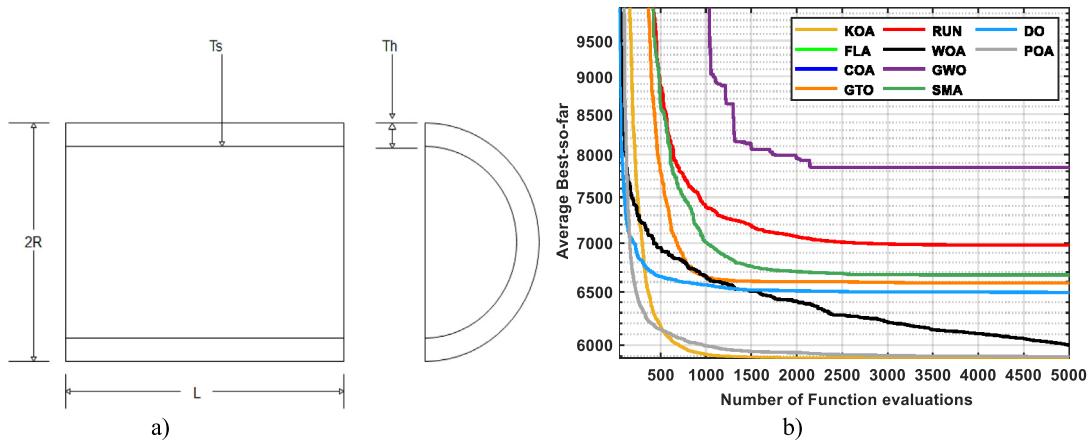


Fig. 15. Pressure Vessel: (a) Schematic; (b) KOA'S Convergence curve.

Table 15
Comparison using Tension/compression spring design.

Algorithms	x_1	x_2	x_3	Best	Avg	SD	Rank	p-value
KOA	0.051689	0.356715	11.289151	0.0126652328	0.012665	6.00E-08	1	
FLA	0.051694	0.356214	11.353478	0.0127111501	30.01109	4.66E+01	8	2.800E-11
COA	0.050000	0.311370	14.862261	0.0131260069	70.00555	4.66E+01	11	1.102E-11
GTO	0.051172	0.344416	12.048357	0.0126701469	13.34504	3.46E+01	6	3.001E-11
RUN	0.053107	0.391807	9.493688	0.0127011107	0.014286	1.63E-03	3	3.020E-11
GWO	0.051243	0.346057	11.945263	0.0126720202	16.67767	3.79E+01	7	2.982E-11
WOA	0.053223	0.394756	9.363685	0.0127071382	0.014825	1.25E-03	4	3.020E-11
SMA	0.051878	0.361284	11.026277	0.0126660030	40.00888	4.98E+01	10	2.521E-11
DO	0.051829	0.360087	11.094154	0.0126656038	0.013086	5.45E-04	2	3.020E-11
POA	0.051839	0.360342	11.079634	0.0126656466	33.34268	4.79E+01	9	2.722E-11

Bold values stands for the best findings.

Table 16
Comparison using Pressure Vessel Design problem.

Algorithms	x_1	x_2	x_3	x_4	Best	Avg	SD	Rank	p-value
KOA	0.778179	0.384659	40.319619	200.000000	5885.434175	5885.4342	1.265E-08	1	
FLA	0.962769	0.480582	49.336397	106.167069	6437.549483	14438.8671	1.380E+04	9	3.0199E-11
COA	1.087684	0.566476	51.651887	85.784256	7220.564900	59599.7164	5.859E+04	11	3.0199E-11
GTO	0.778179	0.384659	40.319619	200.000000	5885.434175	6591.6124	6.705E+02	5	3.0199E-11
RUN	0.778184	0.384672	40.319672	200.000000	5885.517379	6976.0586	5.466E+02	7	3.0199E-11
GWO	0.778560	0.385243	40.335902	199.790300	5888.060222	5997.3776	3.323E+02	3	5.5605E-10
WOA	0.785407	0.498585	40.463169	198.011280	6250.079295	7846.2415	1.172E+03	8	3.0199E-11
SMA	0.778295	0.384717	40.325669	199.915795	5885.634064	6670.7855	5.382E+02	6	3.0123E-11
DO	0.778206	0.384694	40.321004	199.980729	5885.549127	6496.9347	5.602E+02	4	3.0199E-11
POA	0.778179	0.384659	40.319619	200.000000	5885.434175	5895.7232	3.098E+01	2	3.0199E-11

Bold values refer to the best findings.

6.4. The three-bar truss design

$$g_3(X) = -\pi x_4 x_3^2 - \frac{4}{3} x_3^3 + 1296000 \leq 0$$

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

$$\begin{aligned} \text{Variable range } & 0 \leq x_1 \leq 99 \\ & 0 \leq x_2 \leq 99 \\ & 10 \leq x_3 \leq 200 \\ & 10 \leq x_4 \leq 200 \end{aligned}$$

According to Table 16, KOA could be the best in terms of Avg, SD, Rank, and p-value for the pressure vessel design problem. Fig. 15(b) displays the averaged convergence curve of various optimizers, proving that KOA converges better than all.

The objective of this challenge is to discover the ideal values for three variables, including the areas of bars 1, 2, and 3, in order to reduce the weight of a three-bar truss while adhering to specific constraints. Fig. 16(a) displays the problem's general structure [115,116]. The mathematical formulation of this problem is as follows:

- Solution representation:

$$X = [x_1 \ x_2]$$

- Objective function:

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

- Formulation of constraints:

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

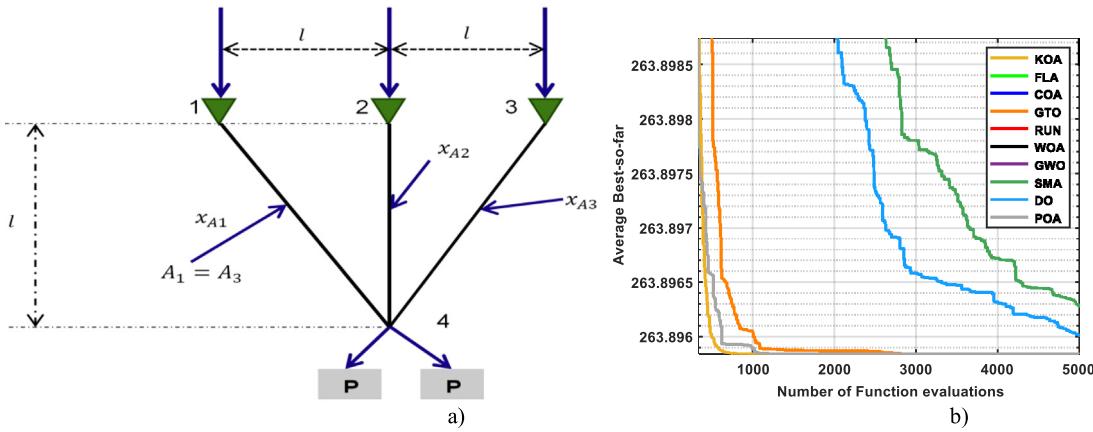


Fig. 16. Three-bar truss problem: (a) Schematic; (b) KOA's Convergence curve.

Table 17

Comparison using three-bar truss design problem.

Algorithms	x_1	x_2	Best	Avg	SD	Rank	p-value
KOA	0.788675	0.408248	263.895843	263.89584	0.00E+00	1	
FLA	0.782254	0.426772	263.932062	267.10146	3.97E+00	9	1.2118E-12
COA	0.788057	0.410073	263.903379	263.96923	7.05E-02	7	1.2118E-12
GTO	0.788675	0.408248	263.895843	263.89584	5.26E-07	2	1.2118E-12
RUN	0.788793	0.407916	263.895854	263.90671	1.41E-02	6	1.2118E-12
GWO	0.788991	0.407357	263.896094	263.89838	2.35E-03	5	1.2118E-12
WOA	0.787151	0.412577	263.897560	264.38526	6.68E-01	8	1.2118E-12
SMA	0.788541	0.408627	263.895857	263.89627	6.75E-04	4	1.2118E-12
DO	0.788643	0.408339	263.895844	263.89601	2.83E-04	3	1.2118E-12
POA	0.788675	0.408248	263.895843	263.89584	0.00E+00	1	NaN

Bold values refer to the best findings.

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

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

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

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

Table 17 compares KOA's findings to this problem with those of many competitive optimizers. It appears from this table that KOA is competitive with POA, and superior to the others for all statistical information. The averaged convergence curves of the proposed KOA and the rival optimizers for this problem are shown in Fig. 16(b); this figure shows that KOA is more converged than all.

6.5. The 10-bar truss design

The final engineering design problem investigated in this paper is the 10-bar truss design problem that is comprised of 10 different design variables, each of which requires accurate optimization in order to get the very best design. In order for this optimal design to be declared finished, it needs to be subjected to ten different stress constraints and twelve different displacement constraints. More information about this problem is available in [113].

Both KOA and the rival optimizers are applied to this problem, and their outcomes are analyzed and reported in Table 18. These findings include the best estimated design variables in addition to the best, Avg, rank, and SD attained by each method across 30 independent runs. Analyzing this table demonstrates that the results produced by KOA are much superior to those produced

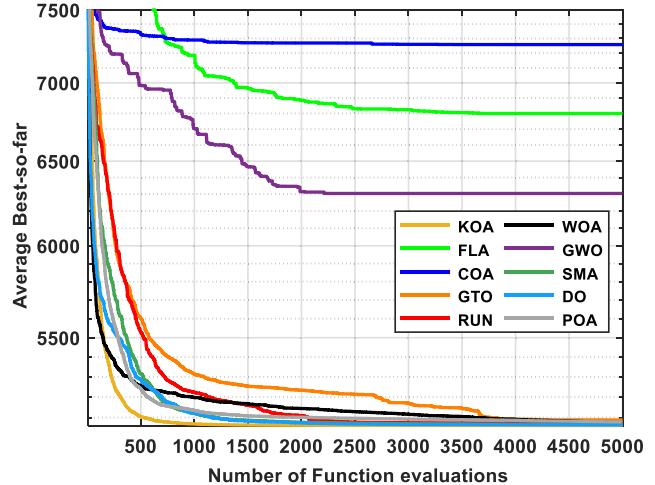


Fig. 17. KOA's Convergence curve over 10-bar truss.

by any of the other algorithms. Even though KOA might perform better than the others in terms of the various statistical information that is being taken into consideration, it is still necessary to assess how well the proposed algorithm performs in terms of the maximum number of function evaluations that are necessary in order to reach the desired average fitness value. As a result of this, Fig. 17 is presented to depict the averaged convergence curves of KOA and the others, which demonstrates that KOA has faster convergence speed.

Table 18

Results for 10-bar truss design problem.

Algorithms	KOA	FLA	COA	GTO	RUN	GWO	WOA	SMA	DO	POA
x_1	30.5231	30.6558	22.3495	30.5036	30.6399	30.6584	27.6049	30.5411	30.7292	30.7129
x_2	0.1000	3.8542	9.4709	0.1001	0.1000	0.1219	0.1000	0.1000	0.1001	0.1005
x_3	23.2092	18.3471	29.1755	22.8213	23.8232	23.1609	19.8560	23.0634	22.9256	23.0085
x_4	15.2430	20.8503	12.1824	15.0435	14.8318	15.2320	18.5056	15.3097	15.0898	14.8720
x_5	0.1000	3.7906	4.8044	0.1000	0.1000	0.1005	0.1276	0.1000	0.1000	0.1000
x_6	0.5504	4.4807	14.1318	0.5254	0.5261	0.5274	2.4688	0.5509	0.5913	0.5638
x_7	21.0264	28.8945	26.5711	21.4145	21.1846	20.6896	22.0021	21.1142	21.0268	21.0640
x_8	7.4552	7.2801	18.7458	7.5527	7.4381	7.4228	7.6459	7.4806	7.4854	7.4837
x_9	0.1000	2.9086	9.7511	0.1000	0.1000	0.1003	0.1000	0.1001	0.1001	0.1004
x_{10}	21.5195	20.9654	11.3411	21.5026	21.1945	21.8952	23.3500	21.4518	21.6303	21.7266
Best	5060.8555	6008.4075	6697.1197	5061.9783	5062.1233	5063.9127	5175.1719	5060.9977	5061.2298	5061.5395
Avg	5060.9321	6799.559	7257.969	5088.474	5073.267	5075.251	6304.3439	5066.867	5064.6542	5078.937
SD	4.20E-02	4.57E+02	3.07E+02	5.26E+01	6.50E+00	8.91E+00	8.03E+02	7.52E+00	5.82E+00	1.82E+01
Rank	1	9	10	7	4	5	8	3	2	6
p-value	3.02E-11	3.02E-11								

Bold values are the best findings.

6.6. Tabular column design

The goal of this challenge is to minimize the cost of designing a tubular column to support a compressive load P under six constraints, as shown in Fig. 18(a). Axial force P influences the column. This problem contains two decision variables: tube thickness T and column diameter d . The column is composed of a material with a yield stress (σ_y) of 500 kgf/cm², an elasticity modulus of 0.85×10^6 kgf/cm², and a density of 0.0025 kgf/cm³. The column's length (L) is 250 cm. The mathematical model of this problem is as follows:

- Solution representation:

$$X = [x_1 \ x_2] = [d \ T]$$

- Objective function:

$$\text{Minimize } f(X) = 9.8x_1x_2 + 2x_1$$

- Formulation of constraints:

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

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

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

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

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

$$g_6(X) = \frac{x_2}{0.8} - 1 \leq 0$$

Variable range : $2 \leq x_1 \leq 14$, $0.2 \leq x_2 \leq 0.8$,

In this section, KOA is used to find a solution to this problem, and its results are compared to those achieved using nine rival optimizers, as shown in Table 19. Based on the data presented in this table, it would appear that KOA is on par with POA and GTO, and superior to the other optimizers. The averaged convergence curves of the proposed algorithm against the other optimizers can be seen in Fig. 18(b), which shows that KOA is a little slower than POA and faster than all the others.

6.7. Cantilever beam design

As illustrated in Fig. 22(a), this challenge consists of five hollow square pieces with the same thickness that is boosted from the first piece, while the remaining pieces are free. The purpose is to reduce the weight of the beam, while the fifth piece of the beam is affected by vertical force. The five pieces' lengths stand in for the five decision variables (x_1, x_2, x_3, x_4, x_5) that are all subject to only one constraint. The mathematical model of this problem is formulated as follows:

- Solutions to this problem are represented as follows:

$$X = [x_1 \ x_2 \ x_3 \ x_4 \ x_5]$$

- Objective function:

$$\text{Minimize } f(X) = 0.0624 \times (x_1 + x_2 + x_3 + x_4 + x_5) \times L$$

- Formulation of constraint:

$$g_1(X) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0$$

Variable range $0.01 \leq x_1, x_2, x_3, x_4, x_5 \leq 100$

The results of KOA and nine competing optimizers are compared in Table 20 to disclose the most effective one to solve the cantilever beam design problem. Based on these results, KOA appears to be superior to the other optimizers. As seen in Fig. 19(b), KOA converges at a higher rate than the other optimizers.

6.8. Gear train design

The challenge of gear train design is an unconstrained optimization and has four integer decision variables: A, B, C, and D, which have to be accurately estimated to minimize the expense of the gear ratio of the gear train depicted in Fig. 20(a). The mathematical formulation is as follows:

- Solution representation:

$$X = [x_1 \ x_2 \ x_3 \ x_4] = [A \ B \ C \ D]$$

- Objective function:

$$\text{Minimize } f(X) = \left(\frac{1}{6.931} - \frac{x_3 x_2}{x_1 x_4} \right)^2$$

Variable range: $12 \leq x_1, x_2, x_3, x_4 \leq 60$

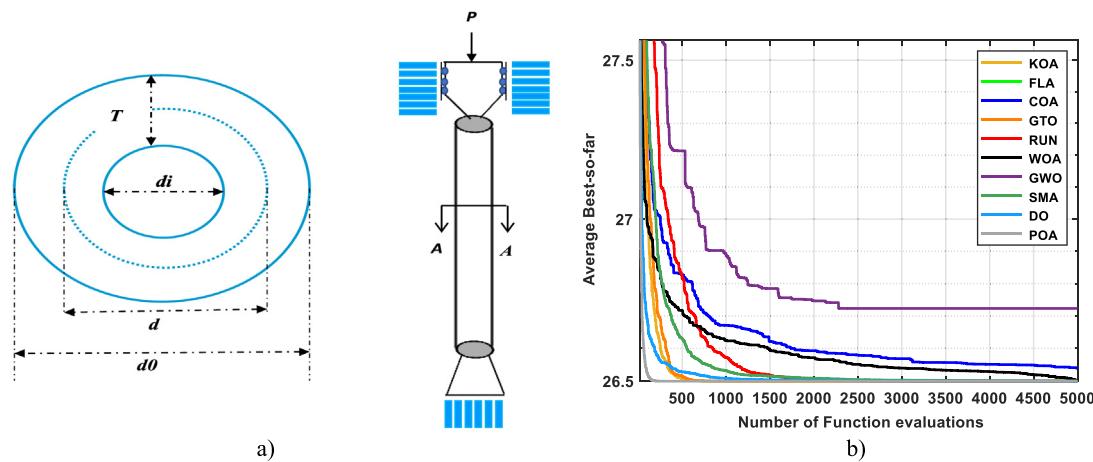


Fig. 18. (a) The schematic of tubular column and A-A cross-section and (b) KOA's Convergence curve.

Table 19

Comparison using tabular column design.

Algorithms	x_1	x_2	Best	Ave	Std	Rank	p-value
KOA	5.4512	0.2920	26.499497	26.499497	1.81E-14	1	
FLA	5.4801	0.2905	26.563266	28.521088	2.08E+00	9	1.2118E-12
COA	5.4511	0.2920	26.501823	26.539035	4.24E-02	6	1.2118E-12
GTO	5.4512	0.2920	26.499497	26.499497	1.81E-14	1	NaN
RUN	5.4512	0.2920	26.499497	26.499502	4.58E-06	3	1.2118E-12
GWO	5.4511	0.2920	26.499770	26.500515	5.02E-04	5	1.2118E-12
WOA	5.4534	0.2918	26.504016	26.723004	2.46E-01	7	1.2118E-12
SMA	5.4512	0.2920	26.499538	26.499839	2.82E-04	4	1.2118E-12
DO	5.4512	0.2920	26.499497	26.499499	1.41E-06	2	1.2118E-12
POA	5.4512	0.2920	26.499497	26.499497	1.81E-14	1	NaN

Bold values are the best findings.

Table 20

Comparison using Cantilever beam design.

Algorithms	x_1	x_2	x_3	x_4	x_5	Best	Ave	Std	Rank	p-value
KOA	6.0160	5.3092	4.4943	3.5015	2.1527	1.339956	1.3399564	4.92E-10	1	
FLA	5.5907	5.5357	4.3654	3.8300	2.3748	1.353859	1.4760899	1.25E-01	10	3.0199E-11
COA	6.5562	5.4120	4.5160	3.1680	2.0082	1.351601	1.4579591	4.74E-02	9	3.0199E-11
GTO	6.0237	5.3041	4.4880	3.5046	2.1534	1.339960	1.340046	1.53E-04	6	3.0199E-11
RUN	6.0155	5.3088	4.4933	3.5041	2.1520	1.339957	1.3399638	8.63E-06	2	3.0199E-11
GWO	6.0122	5.3146	4.4925	3.4984	2.1560	1.339959	1.3400113	3.62E-05	5	3.0199E-11
WOA	5.7240	5.5860	4.6935	3.3631	2.1942	1.345389	1.4233718	5.95E-02	8	3.0199E-11
SMA	6.0336	5.3175	4.4760	3.4991	2.1478	1.339977	1.3400534	7.44E-05	7	3.0199E-11
DO	6.0220	5.3091	4.4932	3.4980	2.1515	1.339958	1.3399644	6.02E-06	3	3.0199E-11
POA	6.0157	5.3088	4.4981	3.4977	2.1534	1.339957	1.3399724	1.11E-05	4	3.0199E-11

Bold values are the best findings.

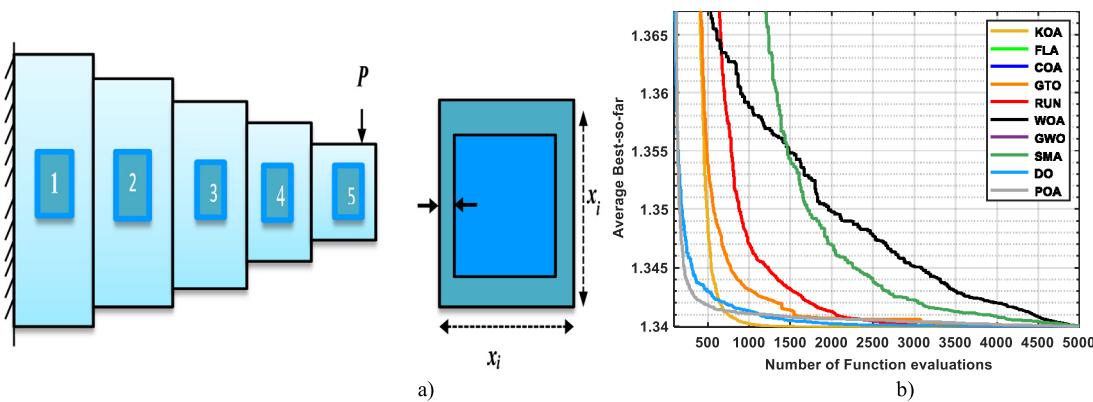


Fig. 19. Cantilever beam design problem: (a) Shape; (b) KPA's Convergence curve.

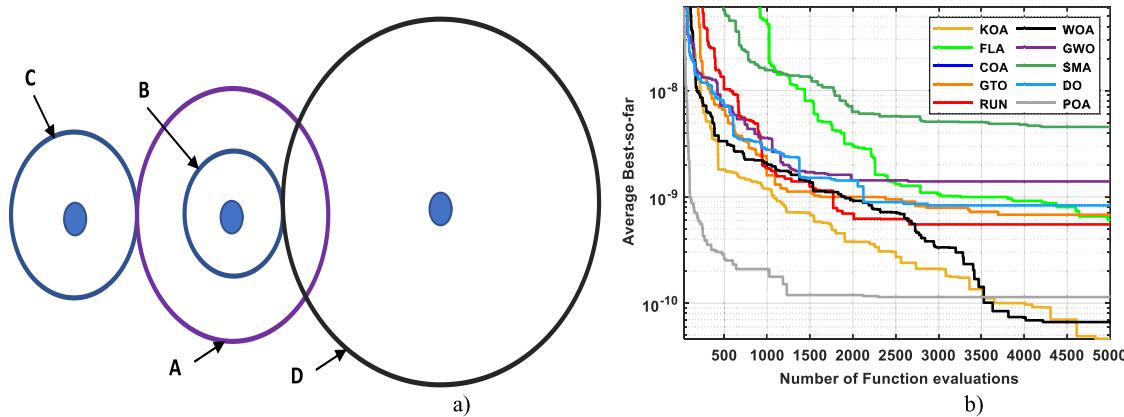


Fig. 20. Gear train design problem: (a) Schematic and (b) KOA's Convergence curve.

Table 21
Comparison using Gear train design.

Algorithms	x_1	x_2	x_3	x_4	Best	Avg	SD	Rank	p-value
KOA	44	20	16	50	2.700857E-12	4.6327E-11	1.6017E-10	1	
FLA	44	16	20	49	2.700857E-12	6.0927E-10	6.6337E-10	5	4.437E-06
COA	23	14	12	48	9.921580E-10	2.0676E-05	1.1228E-04	11	1.589E-11
GTO	50	20	17	44	2.700857E-12	6.7931E-10	6.3293E-10	6	3.640E-07
RUN	44	17	19	49	2.700857E-12	5.5071E-10	6.4104E-10	4	1.806E-06
GWO	44	20	17	49	2.700857E-12	6.6423E-11	1.8386E-10	2	8.309E-02
WOA	50	20	16	44	2.700857E-12	1.3965E-09	3.2918E-09	8	4.946E-07
SMA	52	30	13	53	2.307816E-11	4.5480E-09	8.1244E-09	9	8.442E-10
DO	49	16	19	44	2.700857E-12	8.3176E-10	8.5211E-10	7	1.081E-06
POA	44	17	19	49	2.700857E-12	1.1408E-10	2.7598E-10	3	4.384E-02

Bold values are the best findings.

Table 21 compares the outcomes of utilizing KOA to solve this problem with those obtained using nine rival optimizers. In this table, KOA appears to be on par with some optimizers, and superior to the other optimizers in terms of the best fitness value. **Fig. 21(b)** compares the proposed algorithm's averaged convergence curve to that of existing optimizers, demonstrating that KOA converges at a faster rate than the others.

7. Application II: Parameter estimation of photovoltaic modules

This section evaluates KOA using the single diode model (SDM), the double diode model (DDM), and the triple diode model (TDM) based on the R.T.C France cell. The current–voltage data is measured on a commercial silicon R.T.C. France solar cell with a 57-mm diameter and a temperature of 33 °C [117]. These models include parameters that are not included on the manufacturing sheet and must be precisely adjusted for each model to be implemented accurately and effectively. The lower and upper bounds of these unknown parameters are listed in **Table 22** as being extensively used in published research [118]. The goal of this task is to determine the unknown parameter values that minimize the root mean squared error (RMSE) between the measured and predicted current. This problem's objective function, RMSE, is mathematically expressed as follows:

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

where I_e refers to the estimated current, which is explained in depth in the following subsections. I_m represents the current measurement. M represents the length of measured data. X_i represents the parameters acquired by the i th solution. Based on the provided parameters and Newton–Raphson, I_e is calculated

Table 22
The search boundary of the unknown parameters.

Parameter	R.T.C France cell	
	L_b	U_b
I_{ph} (A)	0	1
I_{sd}, I_{sd1}, I_{sd2} (A)	0	1×10^{-6}
R_s (Ω)	0	0.5
R_{sh} (Ω)	0	100
n, n_1, n_2	1	2

as described in detail in [119]. The controlling parameters of the proposed and rival algorithms for this challenge are set as aforementioned, except N and T_{max} , which were set at 20 and 200,000, respectively, to ensure a fair comparison.

7.1. Single-diode model

Fig. 21 depicts the electrical circuit of the SDM. The SDM output, I , is calculated using the following formula:

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

Where I_{ph} represents the photo-generated current [120]. I_D expresses the diode current and could be computed as follows:

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

where I_{sd} is the reverse saturation current of this diode, V is the output voltage, R_s is the series resistance, n is the diode ideality factor, and V_t is computed as follows:

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

where T is the temperature degree of the junction in kelvin, k indicates the Boltzmann constant ($1.3806503 \times 10^{-23}$ J/K), and q

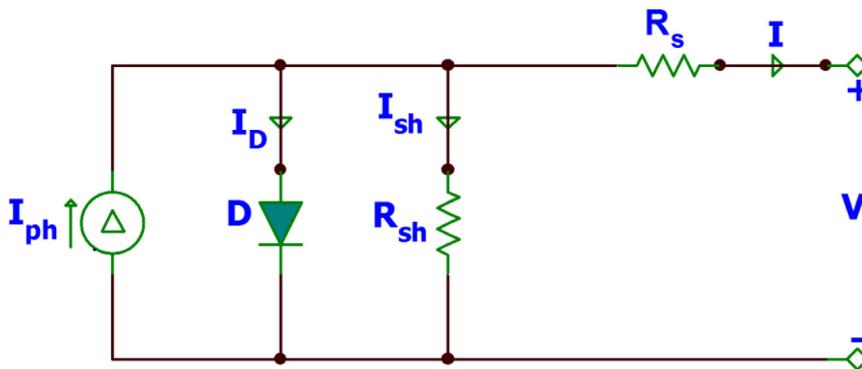


Fig. 21. Equivalent circuit of SDM.

Table 23
Comparison using SDM.

	Best-obtained parameters					RMSE					p-value
	$I_{ph}(A)$	$I_{sd}(\mu A)$	$R_s(\Omega)$	$R_{sh}(\Omega)$	n	Best	Worst	Avg	SD	Rank	
KOA	0.76079	3.1068E-07	0.0365	52.8898	1.4773	0.000773006	0.000773006	0.000773006	1.16065E-17	1	
SO	0.71319	3.8165E-06	0.0280	82.6260	1.7485	0.005466062	0.110358367	0.031656201	0.036354240	8	1.4144E-09
FLA	0.76109	5.9084E-07	0.0348	96.2012	1.5445	0.001908330	0.096028390	0.018276120	0.028008308	7	1.4144E-09
COA	0.76687	1.2907E-05	0.0096	76.3757	1.9757	0.012582886	0.199272080	0.074116843	0.042183962	9	1.4131E-09
GTO	1.48298	5.0000E-05	5.0000	5.2303	3.0000	0.176385040	0.210929815	0.180530512	0.011457168	11	9.5699E-10
RUN	0.70154	1.0914E-05	0.0416	69.2103	1.9971	0.071701671	0.176385040	0.164338459	0.030262621	10	4.7645E-10
GWO	0.75915	2.2227E-07	0.0385	78.1065	1.4440	0.001589646	0.019543895	0.009819133	0.005746251	4	1.4144E-09
WOA	0.76096	5.2739E-07	0.0341	61.2219	1.5324	0.001174357	0.073694490	0.016259104	0.019870936	6	1.4144E-09
SMA	0.76104	1.9579E-07	0.0385	43.7063	1.4324	0.001047420	0.007236174	0.005464847	0.001649766	3	1.4144E-09
DO	0.76061	3.9230E-07	0.0356	60.3199	1.5010	0.000858621	0.005344370	0.003123569	0.001591208	2	1.4144E-09
POA	0.76104	5.2670E-07	0.0341	60.4779	1.5323	0.001181513	0.073694490	0.011590118	0.023453006	5	1.4093E-09

Bold values indicate the best results.

refers to the electron charge ($1.60217646 \times 10^{-19}$ C). I_{sh} stands for the shunt resistor current and is calculated using the following equation:

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

where R_{sh} stands for the shunt resistance. By Substitution, I can be reformulated as:

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

According to the above description, in order to simulate the SDM accurately and effectively, it is necessary to accurately estimate five unknown parameters: I_{ph} , I_{sd} , n , R_s , R_{sh} . This problem is considered an optimization problem and could be tackled using stochastic algorithms like the proposed algorithm. Therefore, the proposed and competing approaches are tested on the SDM's parameter estimation challenge to see which one is most capable of determining the nearly-optimal values of these parameters that minimize the root-mean-squared error (RMSE) between the measured and estimated values. All algorithms have been run 30 times independently, and the fitness values from these runs have been analyzed and published in Table 23. It is clear from this figure that KOA is superior to competing approaches because it is ranked first. Convergence curve and five-number summary between KOA and competing approaches are shown in Fig. 22(a) and (b). It is apparent from these figures that KOA is superior to all other optimizers. Moreover, Fig. 22(c), (d), and (e) depict I-V measured and calculated; and P-V measured and estimated, and error values between the measured and estimated current, respectively. Upon inspection, the measured and calculated values appear to be consistent.

7.2. Double-diode model

Typically, the single diode (SD) model is not a good choice for a range of applications, particularly at low irradiance levels; therefore, the double diode (DD) model is designed to be a strong alternative to SDM for these cases [121]. As depicted in Fig. 23, The DDM comprises from two diodes; the first is a rectifier, while the second is employed to account for the effect of current originating from recombination and the influence of non-idealities in the SC. The DDM output is computed using the formula shown below:

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

I can be replaced with the following detailed equation:

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

where I_{sd1} is the current of the first diode and I_{sd2} is the current of the second diode. n_1 and n_2 are the ideality factors of the diodes. Seven parameters, I_{ph} , I_{sd1} , I_{sd2} , R_s , R_{sh} , n_1 , and n_2 in this formula, are missing from the manufacturing sheet. Without these values, the DDM cannot be accurately replicated. As a result, this section involves an attempt to estimate the near-optimal values for these parameters using the proposed algorithm and competing ones to determine which one is best suited to deal with the problem at hand. The results from each algorithm are reported in Table 24; this table demonstrates that the proposed approach is superior to all others that were compared. Harmonization of the P-V measured and estimated and the I-V measured and estimated are shown in Fig. 24(c) and (d), respectively, while the convergence curve is shown in Fig. 24(a) and a boxplot is shown in Fig. 24(b).

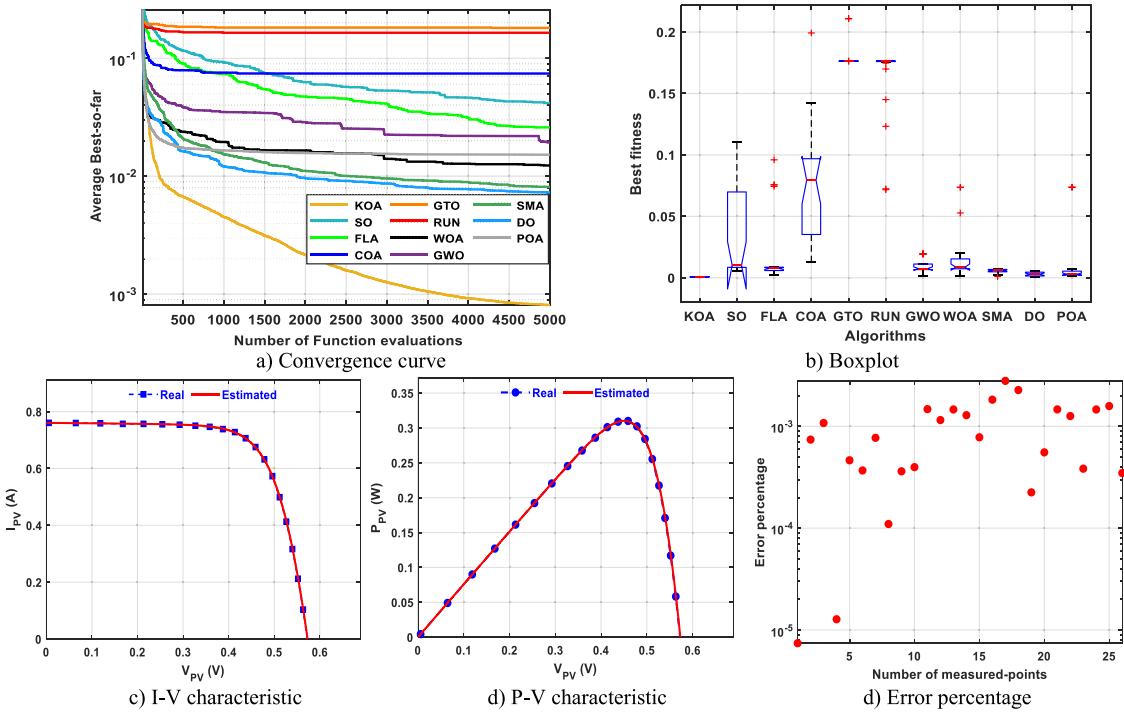


Fig. 22. Various comparisons using SDM.

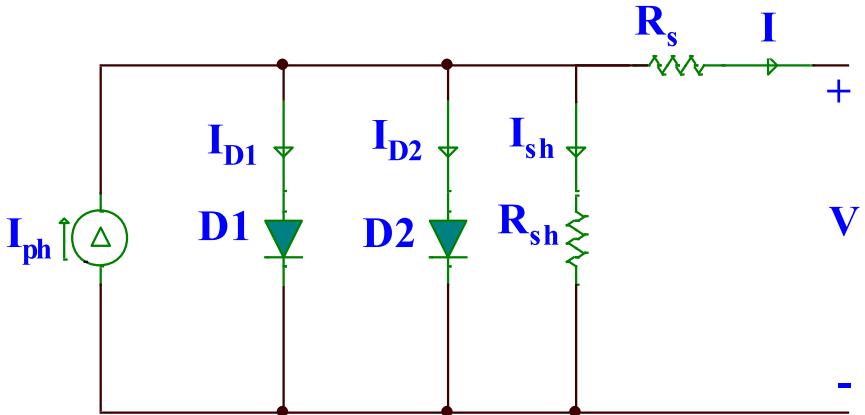


Fig. 23. Electrical circuit of DDM.

In addition, the error values between the measured and estimated current are shown in Fig. 24(d). Those results verify the efficacy of the proposed method.

7.3. Triple-diode model

As depicted in Fig. 25, TDM consists of a source of photo-current (I_{ph}), three parallel diodes, a series resistance (R_s), and a shunt resistance (R_{sh}). The TDM output current is computed using the following formula:

$$I = I_{ph} - \sum_i^3 I_{di} - I_{sh}$$

where I_{di} refers to the i th diode's current [122,123]. I_{di} and I_{sh} can be calculated using the following formula:

$$I_{di} = I_{sdi} \left(e^{\frac{V + IR_s}{a_i V_t}} - 1 \right), \forall i \in 1:3, V_t = \frac{KT}{q}$$

$$I_{sh} = \frac{V + IR_s}{R_{sh}}$$

where I_{sdi} represents the saturation current of the i th diode, and a_i is its ideality factor. To effectively simulate TDM, it is necessary to accurately identify nine unknown parameters, which have been described in detail above as I_{ph} , I_{sdi} , I_{sd2} , I_{sd3} , R_s , R_{sh} , a_1 , a_2 , and a_3 . Therefore, in this section, we offer the results, which result from testing the performance of KOA for estimating such unknown parameters, as well as making comprehensive comparisons with the results of the rival algorithms to show the efficacy of our proposition. Table 25 shows the best retrieved parameters for each algorithm and also reports the fitness value analysis results from 30 separate runs. Based on this table, it is clear that KOA is the most effective method because it outperformed all competitors in terms of all the performance metrics. In addition, Fig. 26 displays the averaged convergence curve, a five-number summary (Boxplot), harmonization between the current I-V and

Table 24

Comparison using DDM.

	Best-obtained parameters						RMSE					p-value
	$I_{ph}(A)$	$I_{sd1}(A)$	$I_{sd2}(A)$	$R_s(\Omega)$	$R_{sh}(\Omega)$	n_1	n_2	Best	Worst	Avg	SD	
KOA	0.761	8.66E-08	2.16E-06	3.80E-02	58.356	1.373	2.000	0.0007326	0.0009635	0.0007449	4.6E-05	1
SO	0.799	5.00E-06	5.00E-06	6.15E-01	5.844	1.614	1.814	0.0028530	0.1955505	0.0544711	5.6E-02	6
FLA	0.759	1.29E-09	6.72E-07	3.44E-02	471.285	1.283	1.562	0.0022651	0.0965936	0.0243546	3.4E-02	5
COA	0.755	4.77E-06	4.97E-06	1.74E-02	498.414	1.866	1.999	0.0228154	0.3794876	0.1683757	9.9E-02	7
GTO	0.734	3.24E-06	4.20E-06	8.47E-02	397.785	1.782	1.948	0.1066940	0.1911547	0.1819978	2.0E-02	9
RUN	0.722	5.00E-06	5.00E-06	1.00E-03	5.230	2.000	3.200	0.1705798	0.1911547	0.1763408	9.4E-03	8
GWO	0.761	2.32E-09	3.60E-06	4.12E-02	79.955	1.137	1.918	0.0009839	0.0206233	0.0042918	4.8E-03	3
WOA	0.759	1.13E-09	1.11E-06	3.09E-02	342.118	1.407	1.618	0.0023627	0.0246510	0.0068243	4.7E-03	4
SMA	0.722	1.00E-09	5.00E-06	1.00E-03	5.230	1.390	1.874	0.0689921	0.3914518	0.2959113	6.7E-02	10
DO	0.761	4.86E-08	8.11E-07	3.82E-02	53.222	1.340	1.736	0.0007615	0.0042863	0.0017407	1.0E-03	2
POA	0.761	1.09E-09	1.13E-06	4.11E-02	60.405	1.108	1.699	0.0008214	0.0694424	0.0127817	2.5E-02	4

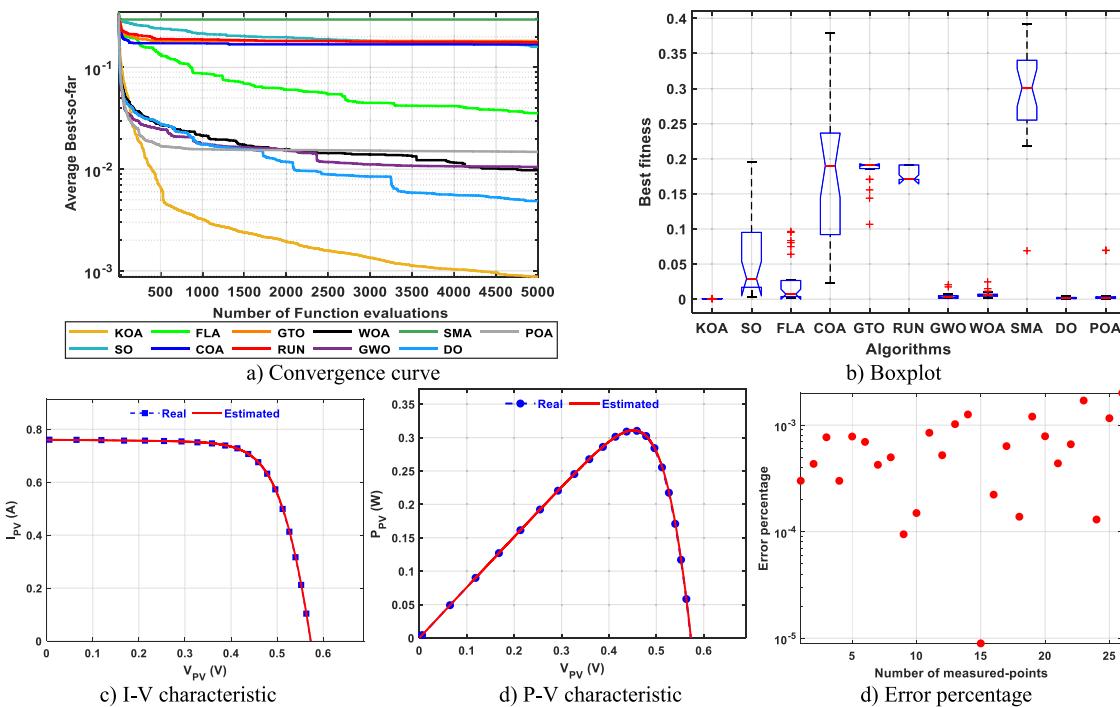
Bold values indicate the best results.

Fig. 24. Various comparisons using DDM.

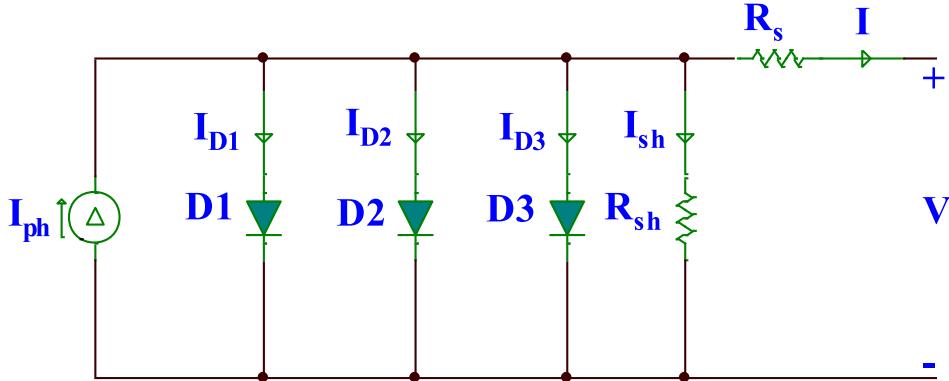


Fig. 25. Electrical circuit of TDM.

the measured, harmonization between the current P-V and the measured, and error values between the measured and estimated

current. Through careful analysis of this figure, we were able to determine that KOA is more effective for this particular challenge.

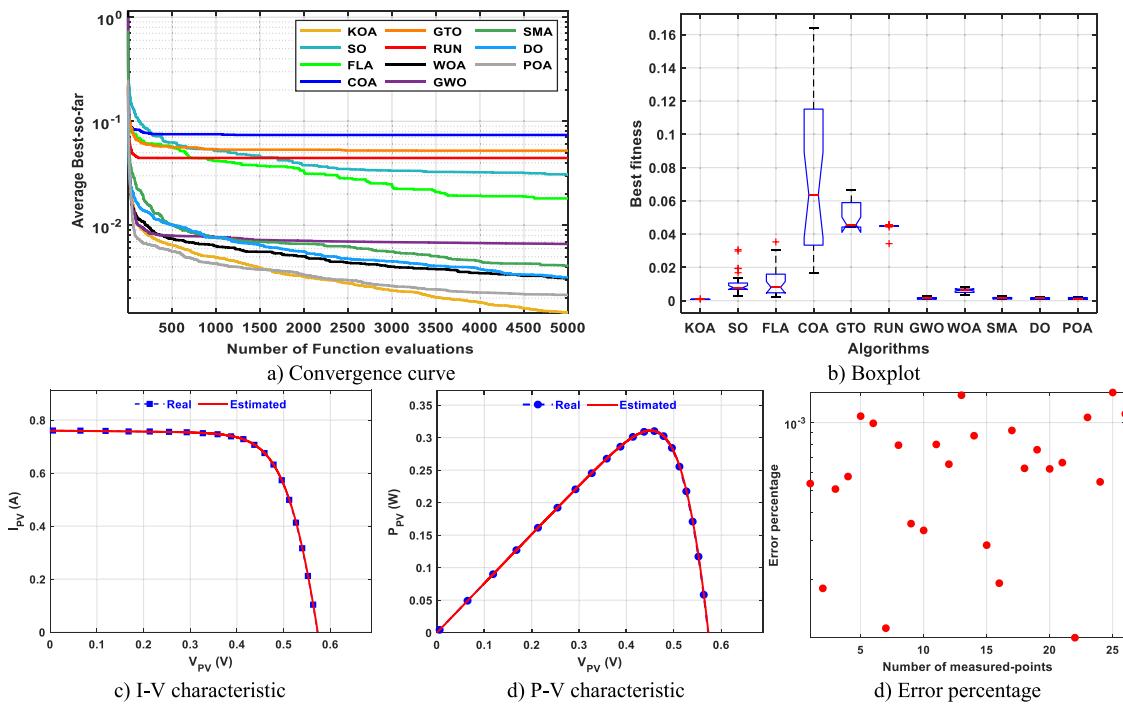


Fig. 26. Various comparisons using TDM.

Table 25

Comparison using TDM.

	Best-obtained parameters									RMSE				p-value	
	$I_{ph}(A)$	$I_{sd1}(A)$	$I_{sd2}(A)$	$I_{sd3}(A)$	$R_s(\Omega)$	$R_{sh}(\Omega)$	n_1	n_2	n_3	Best	Worst	Avg	SD	Rank	
KOA	0.760	2.09E-06	2.79E-08	6.79E-08	3.80E-02	6.13E+01	2.000	1.348	1.400	0.000751	0.001085	0.000791	7.6E-05	1	1.42E-09
SO	0.747	2.75E-07	1.13E-06	1.11E-06	3.40E-02	3.99E+02	1.494	1.737	1.728	0.002680	0.030676	0.010750	7.2E-03	7	1.42E-09
FLA	0.759	4.55E-07	2.17E-06	1.12E-07	3.72E-02	4.82E+02	1.988	1.987	1.400	0.002050	0.035338	0.011191	9.4E-03	8	1.42E-09
COA	0.749	2.58E-06	2.09E-07	7.37E-06	6.87E-03	3.41E+02	1.848	1.853	2.000	0.016430	0.164200	0.073976	4.5E-02	10	1.42E-09
GTO	0.761	1.00E-05	1.00E-09	1.00E-05	1.00E-03	5.23E+00	3.000	1.200	2.000	0.044260	0.066379	0.052264	8.5E-03	11	1.37E-09
RUN	0.732	1.99E-06	4.82E-06	7.80E-07	9.08E-03	6.30E+01	1.914	1.877	1.879	0.034308	0.045653	0.044437	2.1E-03	9	4.72E-10
GWO	0.761	7.35E-08	2.52E-08	2.28E-06	3.83E-02	6.27E+01	1.452	1.309	1.994	0.000755	0.002598	0.001437	5.3E-04	4	2.92E-06
WOA	0.761	2.63E-09	1.07E-06	2.70E-06	2.70E-02	4.09E+02	1.827	1.637	2.000	0.003553	0.008341	0.006037	1.5E-03	6	1.42E-09
SMA	0.760	2.26E-06	1.32E-09	1.06E-07	3.75E-02	6.66E+01	2.000	1.235	1.400	0.000789	0.002600	0.001529	5.1E-04	5	8.28E-09
DO	0.761	1.05E-08	1.70E-06	1.22E-07	3.76E-02	6.00E+01	1.891	2.000	1.400	0.000753	0.002176	0.001369	4.8E-04	3	5.03E-07
POA	0.761	1.49E-07	1.55E-06	1.22E-07	3.76E-02	5.97E+01	1.998	2.000	1.400	0.000753	0.002136	0.001334	5.7E-04	2	4.45E-04

Bold values highlight the best findings.

8. Conclusion and future work

In this study, a novel population-based metaheuristic algorithm, called KOA, is proposed for addressing continuous optimization problems. Its primary inspiration originates from Kepler's laws of planetary motion. The position, mass, gravitational force, and orbital velocity of a planet are four operators that affect the path of planets around the Sun in accordance with Kepler's laws. These operators represent the basis for designing the mathematical model of the proposed KOA. The effectiveness and efficiency of KOA are evaluated utilizing four CEC benchmarks: CEC 2014, CEC 2017, CEC 2020, and CEC 2022. In addition, nine constrained engineering design problems and the parameter estimation problem of photovoltaic modules are used to evaluate KOA further to reveal its ability to handle several real-world optimization problems. Several widely cited, newly proposed, and highly performing rival optimizers are compared with KOA's results to demonstrate that it is more effective in obtaining near-optimal solutions for the vast majority of the studied optimization problems. Although KOA was able to obtain great results for both of the applications that were observed, its performance is still hindered by a little delayed convergence, which is the

primary restriction that will be addressed in subsequent research. In addition, future research will focus on developing binary and multi-objective variants of KOA to assess its performance further.

CRediT authorship contribution statement

Mohamed Abdel-Basset: Investigation, Methodology, Resources, Visualization, Software, Writing – original draft, Writing – review & editing. **Reda Mohamed:** Investigation, Methodology, Resources, Visualization, Software, Writing – original draft, Writing – review & editing. **Shaimaa A. Abdel Azeem:** Conceptualization, Methodology, Writing – review & editing. **Mohammed Jameel:** Conceptualization, Methodology, Writing – review & editing. **Mohamed Abouhawwash:** Conceptualization, Methodology, Resources, Visualization, Validation, Supervision, Writing – review & editing.

Declaration of competing interest

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

Data availability

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

References

- [1] A.A. Heidari, et al., Harris hawks optimization: Algorithm and applications, *Future Gener. Comput. Syst.* 97 (2019) 849–872.
- [2] J. Nocedal, S.J. Wright, *Numerical Optimization*, Springer, 1999.
- [3] G. Wu, Across neighborhood search for numerical optimization, *Inform. Sci.* 329 (2016) 597–618.
- [4] G. Wu, et al., A variable reduction strategy for evolutionary algorithms handling equality constraints, *Appl. Soft Comput.* 37 (2015) 774–786.
- [5] H. Mohammadzadeh, F.S. Gharehchopogh, A multi-agent system based for solving high-dimensional optimization problems: A case study on email spam detection, *Int. J. Commun. Syst.* 34 (3) (2021) e4670.
- [6] S. Mahdavi, M.E. Shiri, S. Rahnamayan, Metaheuristics in large-scale global continues optimization: A survey, *Inform. Sci.* 295 (2015) 407–428.
- [7] M.N. Omidvar, X. Li, X. Yao, A review of population-based metaheuristics for large-scale black-box global optimization—Part I, *IEEE Trans. Evol. Comput.* 26 (5) (2021) 802–822.
- [8] F.S. Gharehchopogh, et al., Advances in sparrow search algorithm: a comprehensive survey, *Arch. Comput. Methods Eng.* 30 (1) (2023) 427–455.
- [9] J. Dréo, et al., *Metaheuristics for Hard Optimization: Methods and Case Studies*, Springer Science & Business Media, 2006.
- [10] Z. Beheshti, S.M.H. Shamsuddin, A review of population-based metaheuristic algorithms, *Int. J. Adv. Soft Comput. Appl.* 5 (1) (2013) 1–35.
- [11] 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) 40.
- [12] F.S. Gharehchopogh, An improved tunicate swarm algorithm with best-random mutation strategy for global optimization problems, *J. Bionic Engineering* 19 (4) (2022) 1177–1202.
- [13] M. Abdel-Basset, R. Mohamed, S. Mirjalili, A novel whale optimization algorithm integrated with Nelder-Mead simplex for multi-objective optimization problems, *Knowl.-Based Syst.* 212 (2021) 106619.
- [14] F.S. Gharehchopogh, Quantum-inspired metaheuristic algorithms: Comprehensive survey and classification, *Artif. Intell. Rev.* (2022) 1–65.
- [15] H. Hakli, H. Uğuz, A novel particle swarm optimization algorithm with levy flight, *Appl. Soft Comput.* 23 (2014) 333–345.
- [16] J. Kennedy, R. Eberhart, Particle swarm optimization, in: *Proceedings of ICNN'95-International Conference on Neural Networks*, IEEE, 1995.
- [17] S. Mirjalili, A. Lewis, The whale optimization algorithm, *Adv. Eng. Softw.* 95 (2016) 51–67.
- [18] J.O. Agushaka, A.E. Ezugwu, L. Abualigah, Dwarf mongoose optimization algorithm, *Comput. Methods Appl. Mech. Engrg.* 391 (2022) 114570.
- [19] S. Saremi, S. Mirjalili, A. Lewis, Grasshopper optimisation algorithm: theory and application, *Adv. Eng. Softw.* 105 (2017) 30–47.
- [20] B. Abdollahzadeh, F. Soleimani 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.
- [21] F. MiarNaeimi, G. Azizyan, M. Rashki, Horse herd optimization algorithm: A nature-inspired algorithm for high-dimensional optimization problems, *Knowl.-Based Syst.* 213 (2021) 106711.
- [22] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, *IEEE Trans. Syst. Man Cybern. B* 26 (1) (1996) 29–41.
- [23] M. Abdel-Basset, et al., Nutcracker optimizer: A novel nature-inspired metaheuristic algorithm for global optimization and engineering design problems, *Knowl.-Based Syst.* (2023) 110248.
- [24] M.S. Braik, Chameleon swarm algorithm: A bio-inspired optimizer for solving engineering design problems, *Expert Syst. Appl.* 174 (2021) 114685.
- [25] S.T. Shishavan, F.S. Gharehchopogh, An improved cuckoo search optimization algorithm with genetic algorithm for community detection in complex networks, *Multimedia Tools Appl.* 81 (18) (2022) 25205–25231.
- [26] F.A. Hashim, et al., Honey badger algorithm: New metaheuristic algorithm for solving optimization problems, *Math. Comput. Simulation* 192 (2022) 84–110.
- [27] M. Zhang, G. Wen, J. Yang, Duck swarm algorithm: a novel swarm intelligence algorithm, 2021, arXiv preprint [arXiv:2112.13508](https://arxiv.org/abs/2112.13508).
- [28] L. Abualigah, et al., Aquila optimizer: a novel meta-heuristic optimization algorithm, *Comput. Ind. Eng.* 157 (2021) 107250.
- [29] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Softw.* 69 (2014) 46–61.
- [30] A. Faramarzi, et al., Marine predators algorithm: A nature-inspired metaheuristic, *Expert Syst. Appl.* 152 (2020) 113377.
- [31] W. Zhao, L. Wang, S. Mirjalili, Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications, *Comput. Methods Appl. Mech. Engrg.* 388 (2022) 114194.
- [32] F.S. Gharehchopogh, An improved harris hawks optimization algorithm with multi-strategy for community detection in social network, *J. Bionic Engineering* (2022) 1–23.
- [33] L. Abualigah, et al., Reptile search algorithm (RSA): A nature-inspired meta-heuristic optimizer, *Expert Syst. Appl.* 191 (2022) 116158.
- [34] S. Li, et al., Slime mould algorithm: A new method for stochastic optimization, *Future Gener. Comput. Syst.* 111 (2020) 300–323.
- [35] Ü. Can, B. Alataş, Physics based metaheuristic algorithms for global optimization, 2015.
- [36] H. Eskandar, et al., Water cycle algorithm—a novel metaheuristic optimization method for solving constrained engineering optimization problems, *Comput. Struct.* 110 (2012) 151–166.
- [37] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: a gravitational search algorithm, *Inform. Sci.* 179 (13) (2009) 2232–2248.
- [38] S. Kirkpatrick, C.D. Gelatt Jr., M.P. Vecchi, Optimization by simulated annealing, *Science* 220 (4598) (1983) 671–680.
- [39] O.K. Erol, I. Eksin, A new optimization method: big bang–big crunch, *Adv. Eng. Softw.* 37 (2) (2006) 106–111.
- [40] H. Shah-Hosseini, Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation, *Int. J. Comput. Sci. Eng.* 6 (1–2) (2011) 132–140.
- [41] R.A. Formato, Central force optimization, *Prog Electromagn Res* 77 (1) (2007) 425–491.
- [42] A. Hatamlou, Black hole: A new heuristic optimization approach for data clustering, *Inform. Sci.* 222 (2013) 175–184.
- [43] A. Kaveh, S. Talatahari, A novel heuristic optimization method: charged system search, *Acta Mech.* 213 (3) (2010) 267–289.
- [44] A. Kaveh, M. Khayatazar, A new meta-heuristic method: ray optimization, *Comput. Struct.* 112 (2012) 283–294.
- [45] F.F. Moghaddam, R.F. Moghaddam, M. Cheriet, Curved space optimization: a random search based on general relativity theory, 2012, arXiv preprint [arXiv:1208.2214](https://arxiv.org/abs/1208.2214).
- [46] L. Xie, J. Zeng, Z. Cui, General framework of artificial physics optimization algorithm, in: *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, IEEE, 2009.
- [47] F.A. Hashim, et al., Henry gas solubility optimization: A novel physics-based algorithm, *Future Gener. Comput. Syst.* 101 (2019) 646–667.
- [48] Y.-T. Hsiao, et al., A novel optimization algorithm: space gravitational optimization, in: *2005 IEEE International Conference on Systems, Man and Cybernetics*, IEEE, 2005.
- [49] A. Faramarzi, et al., Equilibrium optimizer: A novel optimization algorithm, *Knowl.-Based Syst.* 191 (2020) 105190.
- [50] H. Du, X. Wu, J. Zhuang, *Small-World Optimization Algorithm for Function Optimization*, International Conference on Natural Computation, Springer, 2006.
- [51] B. Alatas, ACROA: artificial chemical reaction optimization algorithm for global optimization, *Expert Syst. Appl.* 38 (10) (2011) 13170–13180.
- [52] L. Abualigah, Multi-verse optimizer algorithm: a comprehensive survey of its results, variants, and applications, *Neural Comput. Appl.* 32 (16) (2020) 12381–12401.
- [53] S.I. Birbil, S.-C. Fang, An electromagnetism-like mechanism for global optimization, *J. Global Optim.* 25 (3) (2003) 263–282.
- [54] H. Shah-Hosseini, The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm, *Int. J. Bio-Inspired Comput* 1 (1–2) (2009) 71–79.
- [55] B. Webster, P.J. Bernhard, A local search optimization algorithm based on natural principles of gravitation, in: *Proceedings of the 2003 International Conference on Information and Knowledge Engineering (IKE'03)*, 2003, pp. 23–26.
- [56] B. Javidy, A. Hatamlou, S. Mirjalili, Ions motion algorithm for solving optimization problems, *Appl. Soft Comput.* 32 (2015) 72–79.
- [57] C.-L. Chuang, J.-A. Jiang, Integrated radiation optimization: inspired by the gravitational radiation in the curvature of space-time, in: *2007 IEEE Congress on Evolutionary Computation*, IEEE, 2007.
- [58] P. Rabanal, I. Rodríguez, F. Rubio, Using river formation dynamics to design heuristic algorithms, in: *International Conference on Unconventional Computation*, Springer, 2007.
- [59] M. Abdel-Basset, et al., Young's double-slit experiment optimizer: A novel metaheuristic optimization algorithm for global and constraint optimization problems, *Comput. Methods Appl. Mech. Engrg.* 403 (2023) 115652.
- [60] M. Abdel-Basset, et al., Light spectrum optimizer: A novel physics-inspired metaheuristic optimization algorithm, *Mathematics* 10 (19) (2022) 3466.
- [61] F.A. Hashim, et al., Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems, *Appl. Intell.* 51 (3) (2021) 1531–1551.

- [62] J.H. Holland, Genetic algorithms, *Sci. Am.* 267 (1) (1992) 66–73.
- [63] J.R. Koza, J.R. Koza, *Genetic Programming: On the Programming of Computers By Means of Natural Selection*, vol. 1, MIT Press, 1992.
- [64] H.-G. Beyer, H.-P. Schwefel, Evolution strategies—a comprehensive introduction, *Nat. Comput.* 1 (1) (2002) 3–52.
- [65] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, *IEEE Trans. Evol. Comput.* 3 (2) (1999) 82–102.
- [66] K.V. Price, Differential evolution, in: *Handbook of Optimization*, Springer, 2013, pp. 187–214.
- [67] D. Simon, Biogeography-based optimization, *IEEE Trans. Evol. Comput.* 12 (6) (2008) 702–713.
- [68] R.-J. Kuo, F.E. Zulvia, The gradient evolution algorithm: A new metaheuristic, *Inform. Sci.* 316 (2015) 246–265.
- [69] 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.
- [70] M.S. Kiran, TSA: Tree-seed algorithm for continuous optimization, *Expert Syst. Appl.* 42 (19) (2015) 6686–6698.
- [71] Q. Askari, I. Younas, M. Saeed, Political optimizer: A novel socio-inspired meta-heuristic for global optimization, *Knowl.-Based Syst.* 195 (2020) 105709.
- [72] A. Naik, S.C. Satapathy, Past present future: a new human-based algorithm for stochastic optimization, *Soft Comput.* 25 (20) (2021) 12915–12976.
- [73] Z.W. Geem, J.H. Kim, G.V. Loganathan, A new heuristic optimization algorithm: harmony search, *Simulation* 76 (2) (2001) 60–68.
- [74] 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.
- [75] N. Ghorbani, E. Babaei, Exchange market algorithm, *Appl. Soft Comput.* 19 (2014) 177–187.
- [76] Y. Shi, Brain storm optimization algorithm, in: *International Conference in Swarm Intelligence*, Springer, 2011.
- [77] 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.
- [78] N.A. Golilarz, et al., ORCA optimization algorithm: a new meta-heuristic tool for complex optimization problems, in: *2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing, ICCWAMTIP*, IEEE, 2020.
- [79] H. Jia, X. Peng, C. Lang, Remora optimization algorithm, *Expert Syst. Appl.* 185 (2021) 115665.
- [80] K.M. Ong, P. Ong, C.K. Sia, A carnivorous plant algorithm for solving global optimization problems, *Appl. Soft Comput.* 98 (2021) 106833.
- [81] A. Mohammadi-Balani, et al., Golden eagle optimizer: A nature-inspired metaheuristic algorithm, *Comput. Ind. Eng.* 152 (2021) 107050.
- [82] M. Azizi, S. Talatahari, A.H. Gandomi, Fire hawk optimizer: A novel metaheuristic algorithm, *Artif. Intell. Rev.* (2022) 1–77.
- [83] O.N. Oyelade, et al., Ebola optimization search algorithm: A new nature-inspired metaheuristic optimization algorithm, *IEEE Access* 10 (2022) 16150–16177.
- [84] M. Braik, et al., White shark optimizer: A novel bio-inspired metaheuristic algorithm for global optimization problems, *Knowl.-Based Syst.* 243 (2022) 108457.
- [85] Q. Zhang, et al., Growth optimizer: A powerful metaheuristic algorithm for solving continuous and discrete global optimization problems, *Knowl.-Based Syst.* 261 (2023) 110206.
- [86] P. Trojovský, M. Dehghani, Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications, *Sensors* 22 (3) (2022) 855.
- [87] B. Abdollahzadeh, et al., Mountain gazelle optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems, *Adv. Eng. Softw.* 174 (2022) 103282.
- [88] F.A. Hashim, A.G. Hussien, Snake optimizer: A novel meta-heuristic optimization algorithm, *Knowl.-Based Syst.* 242 (2022) 108320.
- [89] S. Zhao, et al., Dandelion optimizer: A nature-inspired metaheuristic algorithm for engineering applications, *Eng. Appl. Artif. Intell.* 114 (2022) 105075.
- [90] M. Dehghani, et al., Coati optimization algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems, *Knowl.-Based Syst.* 259 (2023) 110011.
- [91] H.-L. Minh, et al., Termite life cycle optimizer, *Expert Syst. Appl.* 213 (2023) 119211.
- [92] F.A. Hashim, et al., Fick's law algorithm: A physical law-based algorithm for numerical optimization, *Knowl.-Based Syst.* 260 (2023) 110146.
- [93] B. Ma, et al., Running city game optimizer: a game-based metaheuristic optimization algorithm for global optimization, *J. Comput. Design Eng* 10 (1) (2022) 65–107.
- [94] O. Kesemen, et al., Artificial locust swarm optimization algorithm, *Soft Comput.* (2022) 1–39.
- [95] P. Trojovský, M. Dehghani, P. Hanuš, Siberian tiger optimization: A new bio-inspired metaheuristic algorithm for solving engineering optimization problems, *IEEE Access* 10 (2022) 132396–132431.
- [96] V. Goodarzimehr, et al., Special relativity search: A novel metaheuristic method based on special relativity physics, *Knowl.-Based Syst.* 257 (2022) 109484.
- [97] L. Wang, et al., Artificial rabbits optimization: A new bio-inspired metaheuristic algorithm for solving engineering optimization problems, *Eng. Appl. Artif. Intell.* 114 (2022) 105082.
- [98] H. Zamani, M.H. Nadimi-Shahraki, A.H. Gandomi, Starling murmuration optimizer: A novel bio-inspired algorithm for global and engineering optimization, *Comput. Methods Appl. Mech. Engrg.* 392 (2022) 114616.
- [99] Y. Baryshev, P. Teerikorpi, *Discovery of Cosmic Fractals*, World Scientific, 2002.
- [100] B. Stephenson, *Kepler's Physical Astronomy*, Princeton University Press, 1994.
- [101] J.L. Russell, Kepler's laws of planetary motion: 1609–1666, *The British J. His. Sci.* 2 (1) (1964) 1–24.
- [102] J.T. Katsikadelis, Derivation of Newton's law of motion from Kepler's laws of planetary motion, *Arch. Appl. Mech.* 88 (1) (2018) 27–38.
- [103] H. Curtis, *Orbital Mechanics for Engineering Students*, Butterworth-Heinemann, 2013.
- [104] R. Malhotra, M. Holman, T. Ito, Chaos and stability of the solar system, *Proc. Natl. Acad. Sci.* 98 (22) (2001) 12342–12343.
- [105] J. Moser, Is the solar system stable?, *Hamilt. Dyn. Syst: A Reprint Selection* 1 (1987) 20.
- [106] D. Halliday, R. Resnick, J. Walker, *Fundamentals of Physics*, Chapters 33–37, John Wiley & Sons, 2010.
- [107] I. Ahmadianfar, et al., RUN beyond the metaphor: an efficient optimization algorithm based on runge kutta method, *Expert Syst. Appl.* 181 (2021) 115079.
- [108] 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.
- [109] 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.
- [110] J.J. Liang, et al., Problem definitions and evaluation criteria for the CEC 2019 special session on multimodal multiobjective optimization, Computational Intelligence Laboratory, Zhengzhou University, 2019.
- [111] 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.
- [112] M. Abdel-Basset, et al., Exponential distribution optimizer (EDO): a novel math-inspired algorithm for global optimization and engineering problems, *Artif. Intell. Rev.* (2023) 1–72.
- [113] C.A.C. Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, *Comput. Methods Appl. Mech. Engrg.* 191 (11–12) (2002) 1245–1287.
- [114] J. Arora, *Introduction To Optimum Design*, Elsevier, 2004.
- [115] M.-Y. Cheng, D. Prayogo, Symbiotic organisms search: a new metaheuristic optimization algorithm, *Comput. Struct.* 139 (2014) 98–112.
- [116] A.H. Gandomi, X.-S. Yang, A.H. Alavi, Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, *Eng. Comput.* 29 (1) (2013) 17–35.
- [117] T. Easwarakhanthan, et al., Nonlinear minimization algorithm for determining the solar cell parameters with microcomputers, *Int. J. Sol. Energy* 4 (1) (1986) 1–12.
- [118] A. Chauhan, S. Prakash, Optimal parameter estimation of solar photovoltaics through nature inspired metaheuristic and hybrid approaches, *IETE J. Res.* (2023) 1–19.
- [119] H. Nunes, et al., A new high performance method for determining the parameters of PV cells and modules based on guaranteed convergence particle swarm optimization, *Appl. Energy* 211 (2018) 774–791.
- [120] Y.T. Tan, D.S. Kirschen, N. Jenkins, A model of PV generation suitable for stability analysis, *IEEE Trans. Energy Convers.* 19 (4) (2004) 748–755.
- [121] A. Askarzadeh, A. Rezazadeh, Parameter identification for solar cell models using harmony search-based algorithms, *Sol. Energy* 86 (11) (2012) 3241–3249.
- [122] J.G. Fossum, F.A. Lindholm, Theory of grain-boundary and intragrain recombination currents in polysilicon pn-junction solar cells, *IEEE Trans. Electron Devices* 27 (4) (1980) 692–700.
- [123] S. Koohi-Kamali, et al., Photovoltaic electricity generator dynamic modeling methods for smart grid applications: A review, *Renew. Sustain. Energy Rev.* 57 (2016) 131–172.