



OPEN

Red-tailed hawk algorithm for numerical optimization and real-world problems

Seydali Ferahtia^{1,2}, Azeddine Houari¹, Hegazy Rezk³, Ali Djeriou², Mohamed Machmoum¹, Saad Motahhir⁴✉ & Mourad Ait-Ahmed¹

This study suggests a new nature-inspired metaheuristic optimization algorithm called the red-tailed hawk algorithm (RTH). As a predator, the red-tailed hawk has a hunting strategy from detecting the prey until the swoop stage. There are three stages during the hunting process. In the high soaring stage, the red-tailed hawk explores the search space and determines the area with the prey location. In the low soaring stage, the red-tailed moves inside the selected area around the prey to choose the best position for the hunt. Then, the red-tailed swings and hits its target in the stooping and swooping stages. The proposed algorithm mimics the prey-hunting method of the red-tailed hawk for solving real-world optimization problems. The performance of the proposed RTH algorithm has been evaluated on three classes of problems. The first class includes three specific kinds of optimization problems: 22 standard benchmark functions, including unimodal, multimodal, and fixed-dimensional multimodal functions, IEEE Congress on Evolutionary Computation 2020 (CEC2020), and IEEE CEC2022. The proposed algorithm is compared with eight recent algorithms to confirm its contribution to solving these problems. The considered algorithms are Farmland Fertility Optimizer (FO), African Vultures Optimization Algorithm (AVOA), Mountain Gazelle Optimizer (MGO), Gorilla Troops Optimizer (GTO), COOT algorithm, Hunger Games Search (HGS), Aquila Optimizer (AO), and Harris Hawks optimization (HHO). The results are compared regarding the accuracy, robustness, and convergence speed. The second class includes seven real-world engineering problems that will be considered to investigate the RTH performance compared to other published results profoundly. Finally, the proton exchange membrane fuel cell (PEMFC) extraction parameters will be performed to evaluate the algorithm with a complex problem. The proposed algorithm will be compared with several published papers to approve its performance. The ultimate results for each class confirm the ability of the proposed RTH algorithm to provide higher performance for most cases. For the first class, the RTH mostly got the optimal solutions for most functions with faster convergence speed. The RTH provided better performance for the second and third classes when resolving the real word engineering problems or extracting the PEMFC parameters.

Optimization algorithms have played a vital role in recent decades in assisting engineers and designers in finding improved solutions for saving time, money, and energy¹. Numerical optimization methods often employ simple and widely used models². Nevertheless, these algorithms require several gradient information to find superior solutions circling a specific point in a small space³. Furthermore, they are sensitive to starting points, mainly when the issues under consideration include multiple local solutions. Inappropriate initial point selection makes searching for the global optimum solution complex and inconsistent⁴. In recent years, many complicated optimization problems have evolved in various disciplines. These problems frequently involve several choices of optimization variables, complex nonlinear constraints, and objective functions⁵. These algorithms have been used for civil engineering⁶, electrical engineering, image processing⁷, medical and biological applications, and others.

As a result, existing numerical approaches cannot address these complicated problems promptly and precisely. On the other hand, nature inspiration can provide concepts to develop artificial intelligence algorithms to solve these complex problems.

¹Institut de Recherche en Énergie Électrique de Nantes Atlantique, IREENA, Nantes University, Saint-Nazaire, France. ²Laboratoire de Génie Electrique, Dept. of Electrical Engineering, University of M'sila, M'sila, Algeria. ³College of Engineering at Wadi Addawaser, Prince Sattam Bin Abdulaziz University, Al-Kharj, Saudi Arabia. ⁴ENSA, University of Sidi Mohamed Ben Abdellah, Fez, Morocco. ✉email: saad.motahhir@usmba.ac.ma

Recently, various nature-inspired metaheuristic algorithms (MAs) have been developed. These MAs mimic the motions of live beings or natural events. These MAs have been elaborated and employed for solving many optimization issues as competitive alternative solvers⁸. However, these MAs have a common drawback; they frequently exhibit an extra sensitivity to adjusting user-defined parameters. Another disadvantage is that MAs may not always reach the global optimum solution⁹. MAs are divided into two types⁸: single solution-based and population-based. A single solution is processed during optimization in the single solution-based type. In contrast, in the population-based type, solutions are developed in each optimization iteration. Population-based MAs initiate the optimization process by producing a set of random individuals. Each of them represents a possible optimal solution. Iteratively, the population will be developed by substituting the actual population with a recently created one based on certain stochastic operators^{10,11}.

Regardless of how diverse these algorithms are, they always have one common characteristic: the searching operations are divided into two phases: exploration and exploitation¹². As a result, during the early stages of the search process, a well-designed optimizer's exploratory behaviors must have an enriched-enough random character to distribute more random solutions effectively¹³. Hence, it enhanced diverse parts of the search space. After the exploration phase, the exploitation phase is carried out. The optimizer accelerates the search process in a narrow area instead of the whole search space by concentrating on the neighborhood of the best-obtained solutions. An efficient optimizer must strike an acceptable and precise balance between the exploration and exploitation phases. Otherwise, the risk of becoming locked in local optima (local solutions) and having immature convergence downsides grows. According to the No Free Lunch (NFL) theorem [36], all the proposed MAs show a comparable average performance when resolving all possible optimization problems. In other words, no algorithm can be considered a general best algorithm. As a result, the NFL theorem promotes creating and developing more new efficient optimizers. Technically, each MA employs unique evolution mechanisms. According to¹⁴, several fundamental concepts can be used to characterize them. Among these concepts:

- *Parallelism* is utilized in population-based algorithms such as SA and HA. Different individuals are sent out at once to complete a task, and the results are compared. According to the comparison, additional ideas are utilized to assess the population evolution or to create new individuals.
- *Acceptance* is utilized in three cases: 1. Admit temporary solutions that weaken objective function due to search space expansion, 2. Handling the cost function's constraints. There are two ways to cope with the constraints. The first way excludes any solutions if they match the start conditions. The second procedure is used if any solution can be allocated a numerical value. In this case, all solutions must participate, and the initial conditions may match infeasible solutions. 3. Adding restrictions to allowed solutions that improve the optimal solution by, at minimum, the limiting level. When comparing values produced from previous calculations, this strategy aids in avoiding numerical issues.
- *Elitism*: the best solution must be kept from one iteration to the next in repeating population-based algorithms. The elitism principle is utilized to accomplish this by retaining the individual who outlined the best solution and utilizing it as a reference for the following iteration or upgrading it if a better solution is located. The notion of elitism may also be applied to many individuals, directing an élite group of them to the next iteration.
- *Selection* is a probability-based method that generates new individuals from a pool of available ones. This approach may incorporate weights into the probabilistic selection process, in which random individuals are chosen to produce new ones.
- *Decay or Reinforcement*: Decay allows for more initial freedom, followed by incremental flexibility constraints. This approach is based on a decremental updating factor (< 1) used at each iteration. Reinforcement is employed similarly in some cases by applying an incremental factor higher (> 1).
- *Immunity* is gained through finding characteristics of solutions that lead to appropriate settings. Immunity prefers solutions with characteristics similar to those attributes.
- *Self-Adaptation* is a process that allows the parameters of algorithms to be updated in response to the progress of optimization.

A novel high-efficiency SA optimization algorithm is suggested in this study to compete with existing optimizers. The basic concept of the suggested optimizer is based on the hunting skills of Red-tailed Hawks, one of the most intelligent birds of prey. The benefits of evolutionary and swarm approaches have been included in the RTH algorithm's design to outperform the existing optimization algorithms. The proposed RTH contains three stages: high soaring, low soaring, and stooping and swooping stages. The high soaring stage is quite similar to evolutionary approaches' search behavior. The RTH algorithm relays and gathers all search points, beginning with the original position and progressing to the best position. The second and third phases imitate SAs's behavior while moving to the best position, using the information of the preceding site for each point.

Despite the availability of metaheuristic algorithms, there is a continuous need for developing and creating new algorithms that may provide better performance for such problems than the actual ones. The NFL theory confirms this need for new optimizers. To this end, this paper suggests this novel algorithm. The main contribution of this paper is to propose a novel nature-inspired metaheuristic optimization algorithm for better solving optimization issues. The proposed algorithm benefit from the unique hunting strategy of the red-tailed hawk and applies it to tackle various optimization problems. The suggested RTH algorithm's performance has been examined on three optimization problem classes. The first class contains three types of optimization functions:

- Twenty-three standard benchmark functions: unimodal, multimodal, and fixed-dimensional multimodal functions.

- IEEE Congress on Evolutionary Computation 2020 (CEC2020) with 15 and 20 search space dimensions.
- IEEE Congress on Evolutionary Computation 2022 (CEC2022) with 10 and 20 search space dimensions.

The proposed algorithm is compared with eight recent algorithms to confirm its contribution to solving the problems of this class. The considered algorithms in the comparison are Farmland fertility Optimizer (FO)¹⁵, African Vultures Optimization Algorithm (AVOA)¹⁶, Mountain Gazelle Optimizer (MGO)¹⁷, Artificial Gorilla Troops Optimizer (GTO)¹⁸, COOT algorithm¹⁹, Hunger Games Search (HGS)²⁰, Aquila Optimizer (AO)²¹, and Harris Hawks optimization (HHO)²². The findings' accuracy (mean value), robustness (standard deviation), and convergence speed are compared. The second class consists of seven real-world engineering problems that will be thoroughly investigated compared to previously published solutions. These problems include the optimal design of an I-shaped beam, a three-bar truss Design, design of a tubular column, a piston lever, a corrugated design, and design tension/compression spring. The third class provides the parameters extraction for proton exchange membrane fuel cell (PEMFC), which will be used to assess the suggested algorithm with a difficult task. To validate its performance, the proposed algorithm will be compared to numerous published articles for three types of PRMFC: BSC 500W, NedStack PS6, and SR-12 500W.

The rest of this paper is organized as follows: Section “[Related works](#)” presents a related works part that shows and explains a set of related works. Section “[Red-tailed hawk algorithm](#)” reviews the hunting behavior of red-tailed, gives the inspiration source, and describes each stage of the proposed RTH algorithm. This section includes the mathematical model of the proposed algorithm. Section “[Results and discussion](#)” presents the outcomes of RTH in solving the considered problems. This paper ends with a conclusion in Section “[Engineering optimization problems](#)”.

Related works

In recent decades, there has been a rise in the assessment and application of metaheuristic algorithms to tackle optimization problems. In the literature, the population-based MAs may be divided into four categories based on their inspiration^{23,24}: evolutionary (EA), Physics-based (PA), Human-based (HA), and swarm-based MAs (SA). EAs imitate biological evolutionary processes, including recombination, mutation, and selection. The Genetic Algorithm (GA)^{25,26}, Differential Evolution (DE)²⁷, Biogeography-Based Optimizer (BBO)²⁸, and Mind evolutionary algorithm optimization (MEDA)²⁹ are the most well-known EAs. Physical phenomena-inspired algorithms are based on physical laws such as gravity, magnetic force, etc. Gravitational Search Algorithm (GSA)³⁰, Gradient-based optimizer³¹, and Energy Valley Optimizer (EVO)³² are a few examples. Human-based MAs imitate some human activities and behaviors. Socio Evolution and Learning Optimization (SELO)³³, Social Network Search (SNS)³⁴, and Human Felicity Algorithm (HFA)³⁵ are some examples of this category. Swarm-based MAs mimic the social behaviors of animals or organisms living in swarms, communities, or packs³⁶. Particle Swarm Optimization (PSO)³⁷, Salp Swarm Algorithm (SSA)³⁸, and Jellyfish Search algorithm (JSA)³⁹ are the most well-known MAs in this category. The swarm-based MAs have been getting more attention in the last years due to the availability of inspiration sources and their efficacy in resolving various optimization problems. New papers propose or review this category of optimization algorithms, such as Mountain Gazelle Optimizer (MGO)¹⁷, Advances in Spotted Hyena Optimizer⁴⁰, Advances in Tree Seed Algorithm⁴¹, Advances in Sparrow Search Algorithm⁴², Butterfly Optimization Algorithm (BOA)⁴³, Advanced Butterfly Optimization Algorithm (ABOA)⁴⁴, Modified Butterfly Optimization Algorithm with Lagrange Interpolation (MBOALI)⁴⁵, African Vultures Optimization Algorithm (AVOA)¹⁶, Whale optimization algorithm (WOA)⁴⁶, Artificial Gorilla Troops Optimizer (GTO)¹⁸, COOT algorithm¹⁹, Weibull Flight based Moth Flame Optimization (WF-MFO)⁴⁷, Hunger Games Search (HGS)²⁰, Aquila Optimizer (AO)²¹ and so on. On the other hand, Quantum-inspired metaheuristic algorithms developed by combining Quantum Computing (QC) principles into metaheuristic algorithms are gaining more interest⁴⁸. The performance of these algorithms is considerably enhanced by using the QC for boost exploration and exploitation and faster convergence.

Red-tailed hawk algorithm

This part discusses the proposed RTH algorithm. The inspiration source and hunting strategy are discussed in the first subsection. Then, the mathematical model mimicking the red-tailed hawk's behavior is presented, and each stage of the algorithm is analyzed.

Inspiration and behavior during hunting. The red-tailed hawk (*Buteo jamaicensis*) is a bird of prey that breeds over much of North America, from Alaska's interior and northern Canada to Panama and the West Indies. The red-tailed hawk lives in various environments and elevations, such as deserts, grasslands, forests, agricultural fields, and cities. The red-tailed hawk is a predatory, carnivorous eater. Almost every little animal they come upon may be seen as possible prey⁴⁹. Small mammals, such as rodents, are their most common prey, although they also eat birds, fish, reptiles, invertebrates, and amphibians. Prey varies greatly depending on geographical and seasonal availability; however, rodents comprise 85 percent of a hawk's diet⁵⁰.

The red-tailed hawk soars with its wings in a mild dihedral, flapping as little as possible to save energy. Unlike other hawks, the red-tails can fly for long distances thanks to this feature. Because soaring is the most efficient flying mode for these hawks, it is utilized more frequently⁵¹. It moves between 32 and 64 km/h (20–40 mph) when soaring or flapping its wings. The large wings allow the red-tailed to reach 190 km/h (120 mph) when plunging⁵². Red-tailed hawks can fly fast and powerfully while repeatedly diving at perceived threats during nest protection⁵³. As illustrated in Fig. 1, the red-tailed has three types of flying:



Figure 1. Behavior of red-tailed hawk during hunting.

- High soaring (Fig. 1a): It flies highly with its wings in a mild dihedral, flapping as little as possible to save energy to explore the selected area.
- Low soaring (Fig. 1b): after selecting the target position, the red-tailed fly with a low soaring in a spiral movement around the prey. This movement allows it to detect the best location and time to hit the target.
- Stooping and swooping (Fig. 1c): after selecting the best location and moment in the previous step, the red-tailed swooped its prey by stooping and raising its acceleration (from 32–64 to 190 km/h) in a curved direction.

Mathematical model. The suggested red-tailed hawk (RTH) algorithm mimics the red-tailed hawk's hunting behavior. The actions taken at each hunt stage are presented and modeled. This algorithm includes three stages, high soaring, low soaring, and stooping and swooping.

- 1 High soaring: the red-tailed hawk will soar far into the sky, looking for the best location in terms of food availability. Figure 2 illustrates the behavior of the red-tailed hawks during the high soaring stage, and Eq. (1) represents the mathematical model of this stage :

$$X(t) = X_{best} + (X_{mean} - X(t-1)) \cdot Levy(dim) \cdot TF(t) \quad (1)$$

where $X(t)$ represents the red-tailed hawk position at the iteration t , X_{best} is the best-obtained position, X_{mean} is the positions' mean, Levy represents the levy flight distribution function that can be calculated according to Eq. (2), and $TF(t)$ denotes the transition factor function that can be calculated according to Eq. (3).

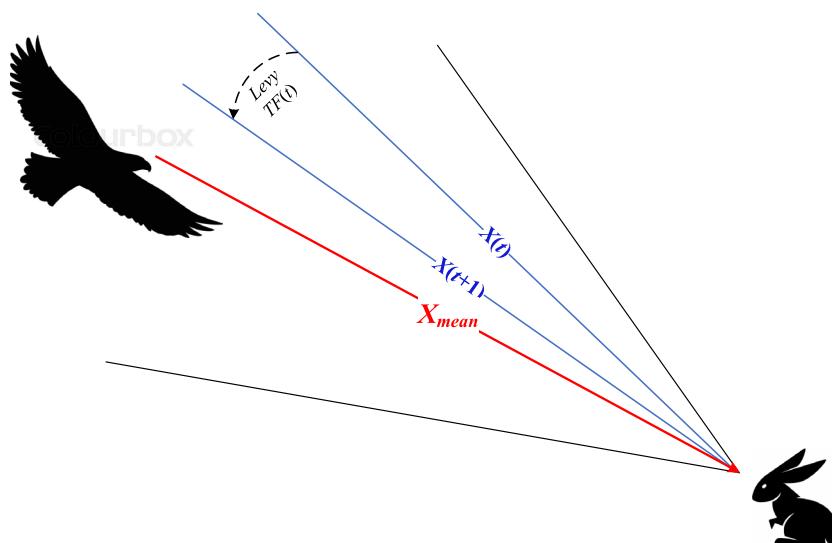


Figure 2. Behavior of red-tailed hawk during high soaring stage.

$$\text{Levy}(\text{dim}) = s \frac{\mu \cdot \sigma}{|v|^{\beta-1}}$$

$$\sigma = \left(\frac{\Gamma(1+\beta) \cdot \sin(\pi\beta/2)}{\Gamma(1+\beta/2) \cdot \beta \cdot 2^{1-\beta/2}} \right) \quad (2)$$

where s is a constant (0.01), dim is the problem dimension, β is a constant (1.5), u , and v are random numbers [0 to 1].

$$TF(t) = 1 + \sin \left(2.5 + \left(t/T_{\max} \right) \right) \quad (3)$$

where T_{\max} represents the max number of iterations.

- 2 Low soaring: the hawk surrounds the prey by flying much lower to the ground in a spiral line. This stage is illustrated in Fig. 3, and its model can be expressed as follows:

$$\begin{aligned} X(t) &= X_{best} + (x(t) + y(t)) \cdot \text{StepSize}(t) \\ \text{StepSize}(t) &= X(t) - X_{mean} \end{aligned} \quad (4)$$

where x and y denote direction coordinates which can be calculated as follows

$$\begin{cases} x(t) = R(t) \cdot \sin(\theta(t)) \\ y(t) = R(t) \cdot \cos(\theta(t)) \end{cases} \quad \begin{cases} R(t) = R_0 \cdot \left(r - t/T_{\max} \right) \cdot \text{rand} \\ \theta(t) = A \cdot \left(1 - t/T_{\max} \right) \cdot \text{rand} \end{cases} \quad \begin{cases} x(t) = x(t)/\max|x(t)| \\ y(t) = y(t)/\max|y(t)| \end{cases} \quad (5)$$

where R_0 represents the initial value of the radius [0.5–3], A denotes the angel gain [5–15], rand is a random gain [0–1], and r is a control gain [1, 2]. These parameters help the hawk fly around the prey with spiral movements, as explained in Fig. 4.

- 3 Stooping and Swooping: In this stage, the hawk suddenly stoops and attacks the prey from the best-obtained position in the low soaring stage. Figure 2 explains the behavior of the red-tailed hawks during this stage. This stage can be modeled as follows:

$$X(t) = \alpha(t) \cdot X_{best} + x(t) \cdot \text{StepSize1}(t) + y(t) \cdot \text{StepSize2}(t) \quad (6)$$

where each step size can be calculated as follows

$$\begin{aligned} \text{StepSize1}(t) &= X(t) - TF(t) \cdot X_{mean} \\ \text{StepSize2}(t) &= G(t) \cdot X(t) - TF(t) \cdot X_{best} \end{aligned} \quad (7)$$

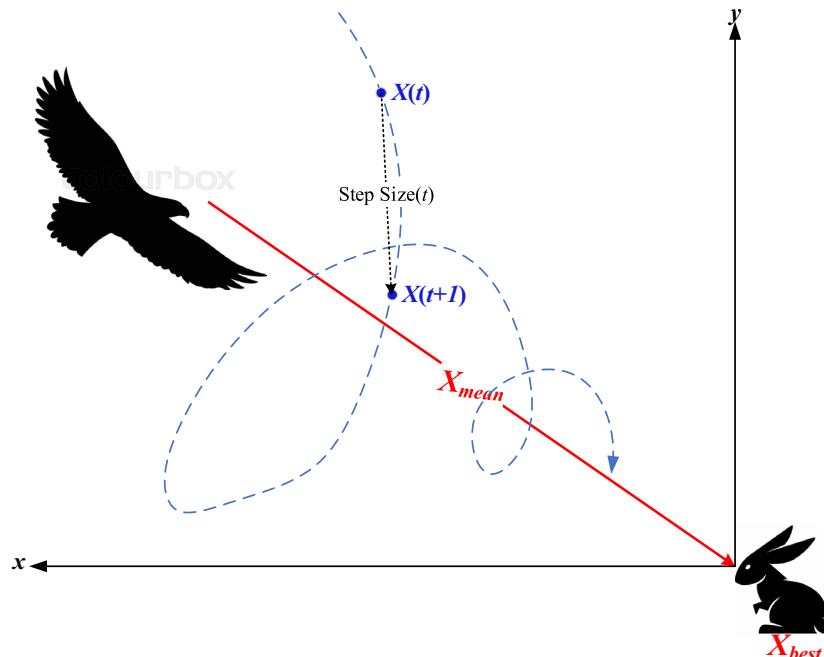


Figure 3. Behavior of red-tailed hawk during low soaring stage.

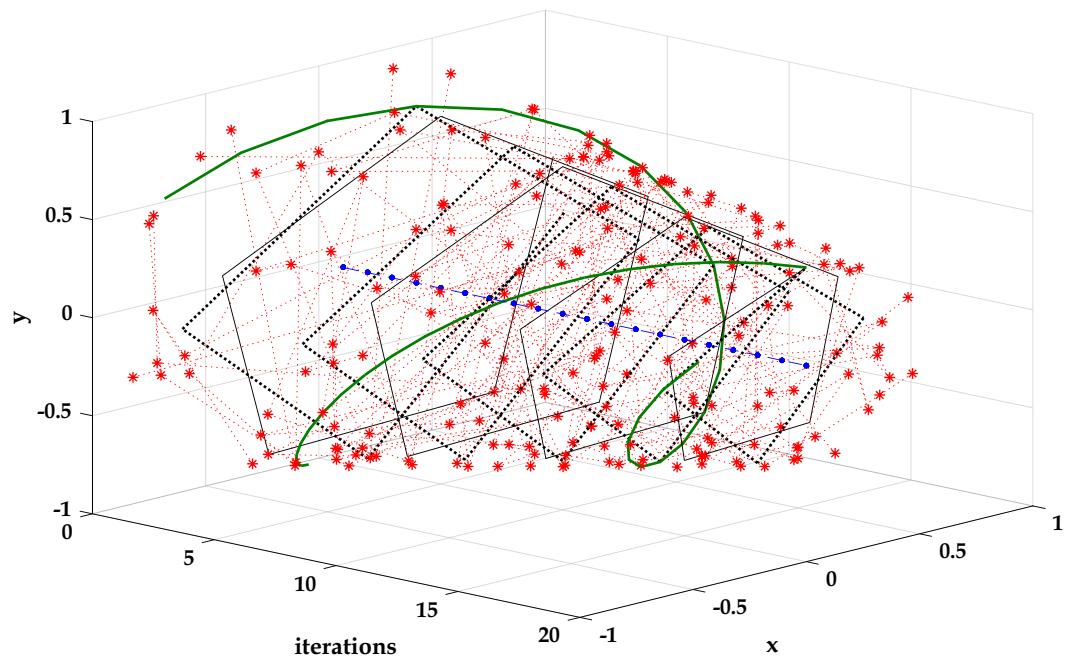


Figure 4. Evolution of the direction coordinates as a function of iterations.

where α and G are the acceleration and the gravity factors, respectively, they can be defined as follows:

$$\begin{aligned}\alpha(t) &= \sin^2\left(2.5 - t/T_{\max}\right) \\ G(t) &= 2 \cdot \left(1 - t/T_{\max}\right)\end{aligned}\quad (8)$$

where α represents the hawk's acceleration that increases with the increase of t to enhance the convergence speed, and G is the gravity effect that decreases to reduce the exploitation diversity when the hawk is much near the prey. This phase is explained in Fig. 5.

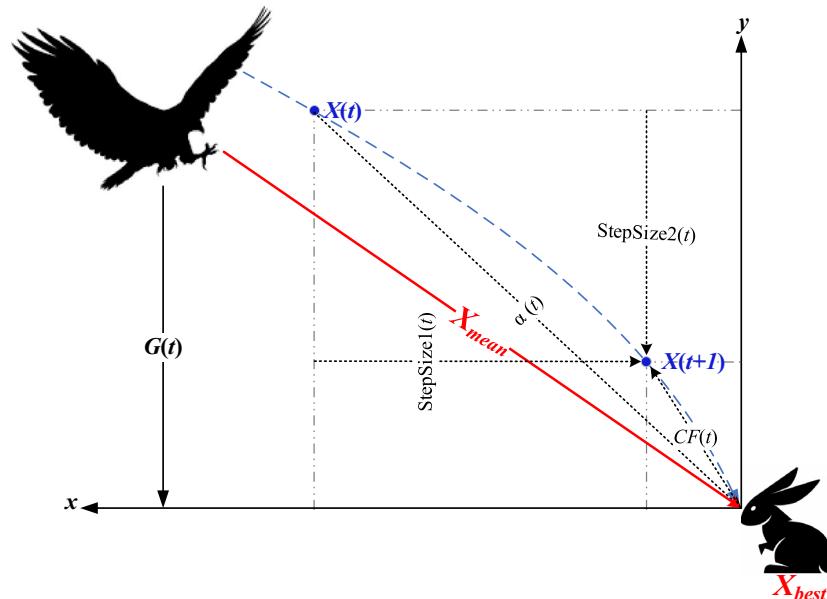


Figure 5. Behavior of red-tailed hawk during stooping and swooping stages.

Figure 6 presents the evolution of the solutions of a 2-dimensional problem as a function of the iterations. During the first iterations, the hawk explored the search space (exploration phase). It took significant steps to detect the prey position. The length of these steps decreases as the hawk is near the prey position. The hawk spent a small number of iterations in the exploration due to the hybrid evolutionary-swarm updating mechanism. It avoided attempting on the local minim thanks to the high soaring stage based on the Levy function. The low soaring enhances the convergence speed in the exploitation phase, where the stooping and the swooping stages strengthen the RTH algorithm's accuracy. Figure 7 presents the hunting behavior for 2-dimensional and 3-dimensional problems. The hawk explores the whole search space from these figures to avoid attempting local

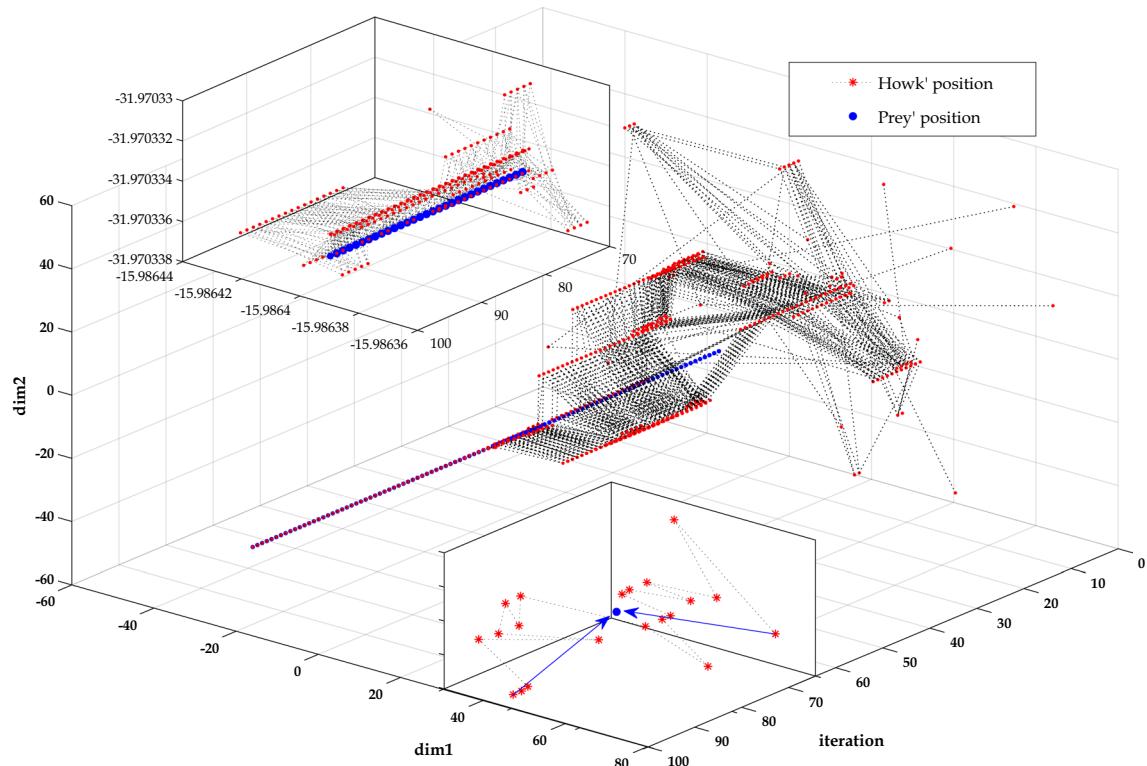


Figure 6. Agents' evolution as a function of the iterations for a 2-dimensional problem.

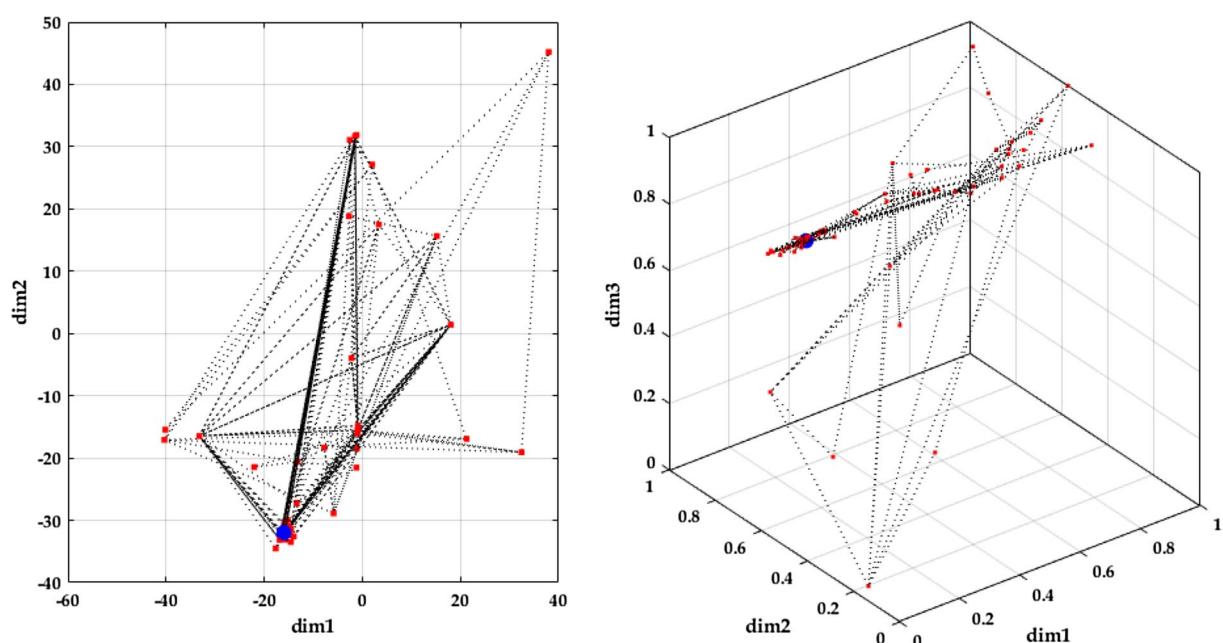


Figure 7. Agents' evolution for 2-dimensional and 3-dimensional problems.

solutions. Due to its unique updating mechanism gets closer to the optimal solution without falling into local ones with a fast convergence rate.

To better understand the characteristics of the proposed RTH and its basics, the RTH concepts are presented in Table 1.

ALGORITHM 1: RTH PSEUDOCODE

```

1   Initialization: random generation within the search space.
2   While  $t < T_{max}$  do
3       High soaring stage: for  $i=1:N_{pop}$  do
4           Calculate Levy flight distribution Eq(2)
5           Calculate the transition factor TF Eq(3)
6           Update positions Eq(1)
7       end
8       Low soaring stage: for  $i=1:N_{pop}$  do
9           Calculate direction coordinates Eq(5)
10          Update positions Eq(4)
11      end
12      Stooping and Swooping stage: for  $i=1:N_{pop}$  do
13          Calculate the acceleration and the gravity factors Eq(8)
14          Calculate the step size Eq(7)
15          Update positions Eq(6)
16      end
17  end

```

Results and discussion

RTH's performance was evaluated using several test functions. Three classes of test functions are used in this study:

- Standard test functions⁵⁴,
- CEC 2020 benchmark test functions⁵⁵,
- CEC 2022 benchmark test functions⁵⁶.

Standard test functions. Unimodal, multimodal, and fixed-dimension multimodal test functions are used to evaluate the RTH performance⁵⁴. Unimodal test functions (F1-F7) are used to put an algorithm's exploitation abilities to the test, whereas multimodal test functions (F8-F13) are used to experiment with the algorithm's exploration performance. The fixed dimension test functions (TF14-TF23) demonstrate the algorithm exploration capability⁵⁷. These functions are provided in Table A1 (appendix). For all the tests, the population size for each algorithm has been set at 30 with max iterations of 1000.

This study ran RTH 30 times and reported the findings, including the average and standard deviation values of the best-so-far solutions obtained in each run to provide statistically significant results. The test is carried out for six other algorithms to underline the effectiveness and superiority of RTH to that of different methods: Particle Swarm Optimization (PSO)³⁷ as the best-known optimization method, Salp Swarm Algorithm (SSA)³⁸ as one of the most used algorithms, Hunger Games Search (HGS)²⁰, COOT algorithm¹⁹, Artificial Eco-system Optimizer (AOE)⁴, and Aquila Optimizer (AO)²¹ as recently developed metaheuristics. Table 2 summarizes the parameter setting for the RTH algorithm, where the parameters are set by the try-and-error method. The other algorithms are based on their default parameters. Table 3 shows the results of RTH and the other used algorithms on these standard test functions using the best, average, and standard deviation (StD) values.

Concept	Projection on the algorithm
Classification according to inspiration type	Swarm-based
Classification according to evolution type	Population-based
Parallelism	The algorithm uses several positions and their mean value
Acceptance	Not used due to the low random behavior of the algorithm
Elitism	Used at each phase (X_{best})
Selection	The high soaring phase used this concept based on Levey's distribution
Decay	Used in α and G Eq. (8) to raise the exploration ability and converge toward the best solution
Reinforcement	Used in TF Eq. (3) to enhance the exploitation ability and avoid attempting in local optima
Immunity	Not used
Self-adaptation	Used in TF Eq. (3) and α and G Eq. (8), these parameters are updated according to the current iteration

Table 1. The RTH concepts.

Parameter	Range	Cosen value
A	[5–25]	15
R_0	[0.1–2]	0.5
r	[0.1–2]	1.5

Table 2. The RTH paramters.

Unimodal functions results (F 01–07) because unimodal test functions have just one global optimum solution, they may be used to assess an algorithm's convergence speed (exploitation phase). From Table 3 results and the curves illustrated in Fig. 8 (fnc 05 curves are not included due to the exact similarity in the results), the RTH outperformed all algorithms in all test functions in terms of accuracy and convergence speed. These results demonstrate RTH's capabilities in quick exploitation, which can enable RTH in fast convergence. This capacity is derived from the high soaring phase that uses the adaptive transition factor (TF) and Lévy distribution.

Multimodal functions results (F 08–13) Multimodal test functions include multiple local solutions. Their number grows exponentially as the number of search space dimensions grows (optimization variables). Having more than one optimal is beneficial if the purpose is to test an algorithm's exploration capabilities. As a result, they are well-suited for assessing optimization algorithms' local optima avoidance and exploratory behavior. The obtained results in Table 3 and Fig. 9 demonstrated that RTH has superior exploration capabilities to other methods for five of six functions in terms of finding the optimal solution with the lowest number of iterations. These functions approve the ability of the RTH to handle complicated problems.

Fixed-dimensional multimodal function (F 14–23) similar to the multimodal test functions, these functions include a lot of local optima with more complexity due to the minimization of the negative cost function. Hence, they are suitable for evaluating the accuracy of the algorithm. As illustrated in Fig. 10 and reported in Table 3, the proposed RTH has reached the global optimum for all the considered functions. However, its robustness, represented by the standard deviation (StD), has been reduced to the increased complexity. RTH's exploration is due to its different optimization phases and the acceleration and gravity effects.

IEEE CEC 2020. The IEEE Congress on Evolutionary Computation (CEC) offers yearly benchmark functions to test and evaluate optimization algorithms' ability to identify optimal solutions (minimization problems). This collection of functions is commonly recognized by the year as standard functions established by the IEEE Congress on Evolutionary Computation, for example, CEC 2020 and CEC 2022. These test functions include unimodal, basic multimodal, hybrid, and composition functions. First, the proposed RTH will be compared with the considered algorithms using the ten functions of the IEEE CEC 2020⁵⁵. The characteristics of CEC 2020 functions are presented in Table A2. Their 3D maps are shown in Fig. 11 for the unimodal function (F1), Fig. 12 for the basic functions (F2–F4), and Fig. 13 for the composition functions (F8–F10).

All the algorithms are run 30 times for each function. The results for these functions for D = 15 are presented in Table 4 and. These include the best, the main, and the standard deviation (StD). The fitness evolutions during the optimization process for each function are presented in Fig. 14.

The three last rows show the scores of each algorithm. It can be seen from the L1 row that the proposed RTH can find the optimal solution for five functions from ten, followed by MGO by three functions, then HGS, AVOA, and GTO by two functions, where the other algorithms achieved only one optimal solution. From the penultimate row, the proposed RTG can accomplish the best mean results for five problems, followed by MGO by four, then HGS by two. The last row presents the StD results that confirm the algorithm's robustness. Based on these results, the proposed RTH provided the best results in this criterion four times. The evolution of the average cost function is presented in Fig. 13. From this figure, the proposed RTH has fast convergence speed compared to the other algorithms. This can be explained by the unique exploitation and exploitation strategies of the proposed RTH algorithm.

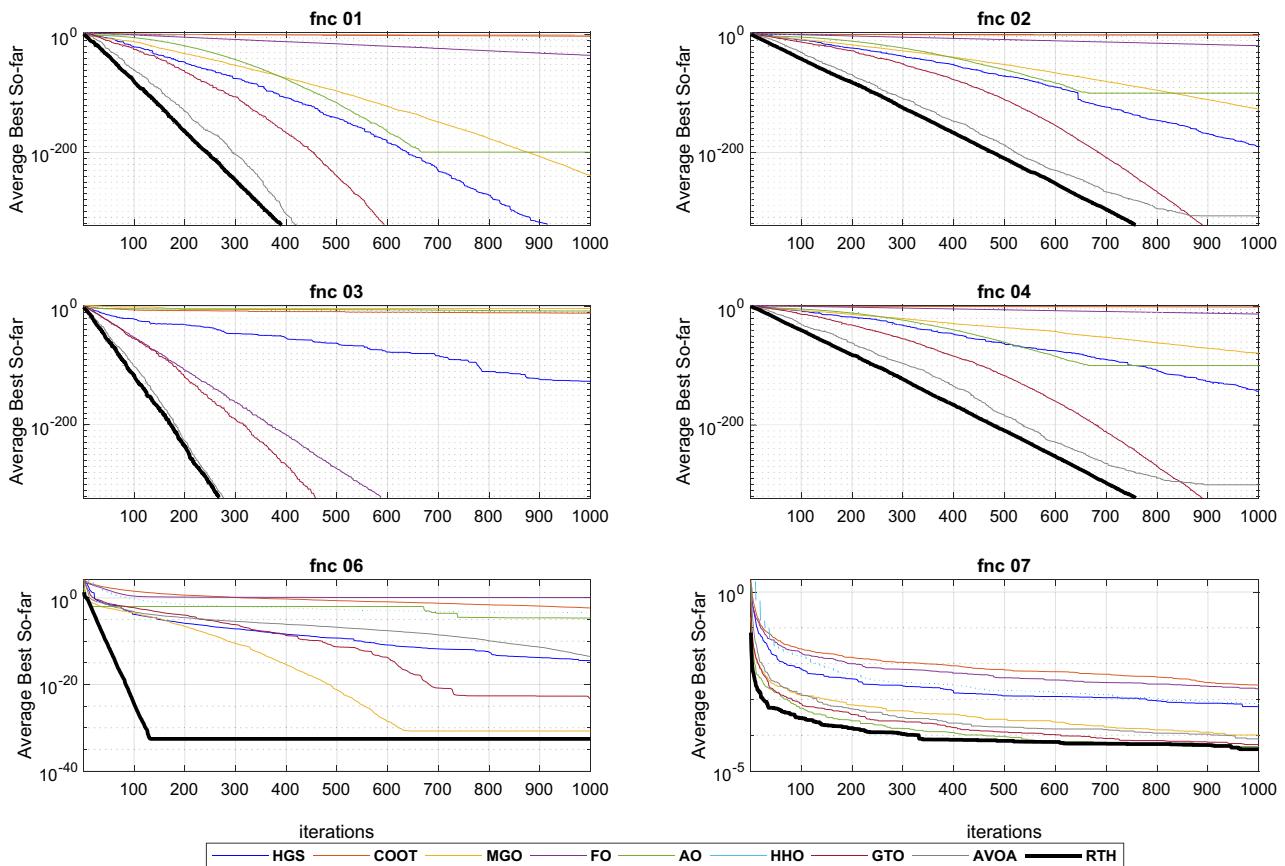
The functions of the CEC 2020 benchmark will be tested with increased search space dimension where D is equal to 20. This allows us to investigate the performance with more complex cases. The results are provided in Table 5.

It can be seen from the L1 row that the proposed RTH can find the optimal solution for six functions, followed by MGO by three, then HGS, AVOA, and GTO by two functions. From the penultimate row, the proposed RTG has got the best mean results for five problems, followed by MGO by three, then HGS by two. From the last row, the StD results of the RTH is the best by five times, which confirms the algorithm's robustness. Based on these results, the proposed RTH provided the best results in this criterion four times. The evolution of the average cost function is presented in Fig. 15. Similar to the previous case, the proposed RTH has fast convergence speed compared to the other algorithms.

ANOVA is the abbreviation for analysis of variance. ANOVA is frequently used to evaluate equality across different means by comparing variance between groups with variation within groups. This test will be used to assess the results between the optimizers for both D = 15 and D = 20 cases. The best outcome for each case is marked in red.

For the unimodal function (function 01), the ANOVA ranking is presented in Fig. 16. From this figure; the RTH generates the best solutions in terms of accuracy with the lowest variation range. On the other hand, the FO and the HHO cannot find any solution.

Func	Metric	MGO	FO	COOT	HGS	AO	HHO	GTO	AVOA	RTH
F18	Best	3	>3	3	3	>3	>3	3	>3	3
	Mean	3	>3	>3	3	>3	>3	3	12.9850033	3.9
	StD	1.9710^{-15}	4.26×10^{-4}	1.22×10^{-8}	3.92×10^{-15}	2.27×10^{-2}	2.144×10^{-5}	8.96×10^{-16}	21.94	4.93
F19	Best	-3.863	-3.854	-3.863	-3.863	-3.863	-3.863	-3.863	-3.863	-3.863
	Mean	-3.863	-3.787	-3.863	-3.863	-3.856	-3.818	-3.863	-3.825	-3.863
	StD	2.58×10^{-15}	0.263	3.20×10^{-9}	2.71×10^{-15}	2.84×10^{-3}	6.94×10^{-2}	2.682×10^{-15}	0.142	2.696×10^{-15}
F20	Best	-3.322	-3.009	-3.322	-3.322	-3.295	-3.187	-3.322	-3.322	-3.322
	Mean	-3.267	-2.751	-3.299	-3.282	-3.208	-2.831	-3.290	-2.888	-3.263
	StD	0.06	0.39	0.05	0.07	0.08	0.33	0.05	0.62	0.06
F21	Best	-10.153	-5.69	-10.153	-10.153	-10.153	-5.055	-10.153	-10.153	-10.153
	Mean	-10.153	-3.221	-9.157	-9.983	-10.149	-5.045	-10.153	-10.153	-7.184
	StD	6.02×10^{-15}	1.278	2.58	0.93	0.01	0.02	6.68×10^{-15}	5.55×10^{-4}	2.68
F22	Best	-10.403	-6.306	-10.403	-10.403	-10.403	-5.088	-10.403	-10.403	-10.403
	Mean	-10.403	-2.917	-9.541	-10.403	-10.399	-5.068	-10.403	-10.224	-6.073
	StD	6.60×10^{-16}	1.20	2.28	7.34×10^{-15}	0.01	0.05	9.33×10^{-16}	0.97	2.790
F23	Best	-10.536	-4.055	-10.536	-10.536	-10.536	-5.129	-10.536	-10.536	-10.536
	Mean	-10.536	-2.838	-10.536	-10.356	-10.533	-5.119	-10.536	-10.236	-6.210
	StD	1.32×10^{-15}	0.61	7.71×10^{-6}	0.99	0.01	0.01	9.33×10^{-16}	1.166	2.81
Results	Best	15	5	10	19	11	9	21	19	22
	Mean	14	3	6	13	4	5	20	9	16
	StD	10	3	2	7	3	2	14	7	10

Table 3. standard test functions results. The best results are marked in bold.**Figure 8.** Convergence curves of the unimodal functions.

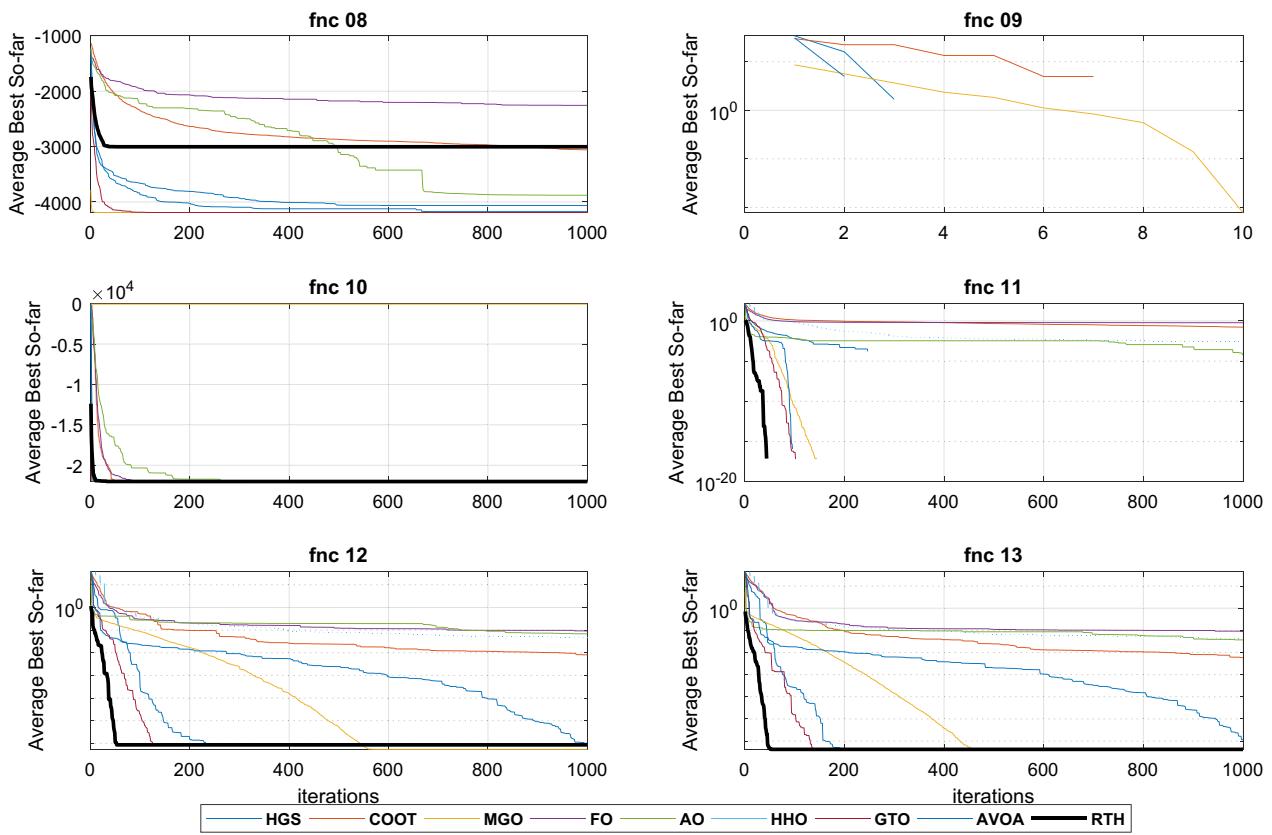


Figure 9. Convergence curves of the multimodal functions.

For the basic functions (fnc 02–04), their results are presented in Fig. 17. For fnc 02 for both cases ($D = 15$ and $D = 20$), all the optimizers except for the F and the HHO got the optimal solution where the COOT provided the lower variation range and the MGO include the set of the best result. The results of function 03 are similar to those of function 02, with slight superiority to the HGS in precision and stability. For function 04, the RTH provided identical results for both cases ($D = 15$ and $D = 20$). This confirms its ability to resolve this problem effectively.

Hybrid functions (fnc 05–07) ANOVA results are presented in Fig. 18. The results of function 06 are not included because of the exact similarity of the results. These figures confirm the outperformance of the proposed RTH for the three functions compared to the other considered algorithms. The MGO and GTO algorithms performed the second best after the RTH.

Composition functions (fnc 08–10) ANOVA results are presented in Fig. 19. For function 08; the MGO provided the best performance where the proposed RTH for the best result, but its variation range is higher due to some ultimate results far from the best. The performance of the algorithms is much closer to each other except for the FO and the HHO for function 09, with light superiority for the MGO. The RTH provides more stable performance and higher accuracy for function 10, followed by the MGO.

IEEE CEC 2022. In this case, the proposed RTH will be compared with the considered algorithms using the 12 functions of the IEEE CEC 2022⁵⁶. The characteristics of CEC 2022 functions are presented in Table A3. Their 3D maps are illustrated in Fig. 20 for the unimodal function (F1), Fig. 21 for the basic functions (F2–F5), and Fig. 22 for the composition functions (F9–F12).

Similar to the CEC 2020 cases, all the algorithms are run 30 times for each function for $D = 10$ and $D = 20$. The obtained results for these functions for $D = 21$ are presented in Table 6.

From the L1 row, the proposed RTH can find the optimal solution for eight from twelve functions, followed by GTO by six functions, then the HGS by five functions, then AVOA by four. This can approve its ability to get the optimal solution and escape from the local solutions. In comparison, FO, AO, and HHO cannot provide any best solution. From the penultimate row, the MGO achieved the best mean results by six functions, followed by the proposed RTH by five functions, then the HGS and GTO by two. This approves the accuracy of the proposed RTH. FO, COOT, AO, HHO, and AVOA algorithms didn't get any mean best results for all the tests. From the last row, the proposed RTH provided the best results by five functions, followed by the proposed MGO by four times. This can approve its robustness. Figure 23 presents the fitness evolution for each function of the CEC 2022 with $D = 10$. These figures support the findings of the CEC 2020, where the proposed RTH gives an excellent convergence speed.

For $D = 20$ case, the results are provided in Table 7.

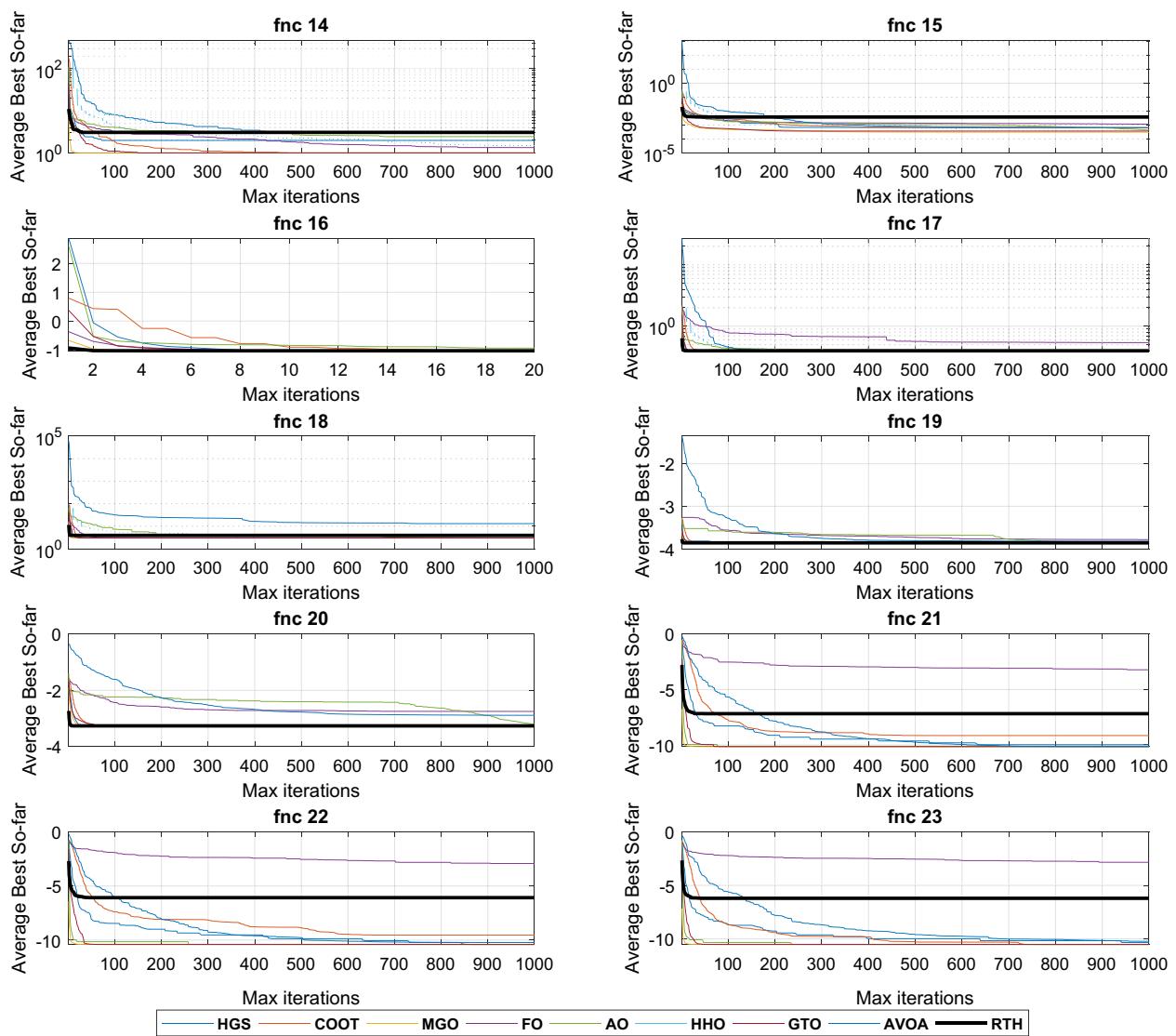


Figure 10. Convergence curves of the multimodal functions.

From the L1, the proposed RTH can find the optimal solution for six from twelve functions, followed by GTO by only three functions, then the HGS by two functions. The increased complexity can explain the decreased number of optimal solutions due to increased search space dimensions. These results approve the RTH's ability to get the optimal solution better and escape from the local solutions even in complicated problems. From L2, the MGO achieved the best mean results by five functions, followed by the proposed RTH by four. This approves the accuracy of the proposed RTH compared to the MGO algorithm. From L3, the proposed RTH provided the best results by five functions, followed by the proposed MGO by three times. This can approve its robustness. Compared to D = 10, the RTH can generally handle problems that include optimization variables better. Figure 24 presents the fitness evolution for each function of the CEC 2022 with D = 20. These figures approve the findings of the CEC 2020 and CEC 2022 with D = 10, where the proposed RTH presents a faster convergence speed.

Figure 25 shows the ANOVA ranking results for unimodal functions for both cases. From this figure, the RTH and the MGO generate the best solutions in terms of accuracy with the lowest variation range for D = 10. However, when the search space dimensions increased, the performance of the MGO decreased where the RTH performance was constant.

The basic functions (fnc 02–05) results are presented in Fig. 26. For fnc 02 for both cases (D = 15 and D = 20), all the optimizers except for the FO and the HHO got the optimal solution where the MGO and RTH provided the lower variation range for both cases. The results of fnc 03 show that the proposed RTH cannot resolve this problem effectively. The MGO and the HGS provide the best performance for this case. For fnc 04, all the optimizers provided near performance with a slight advantage to the MGO. The same comments can be extracted for fnc 05.

Hybrid functions (fnc 06–08) ANOVA results are presented in Fig. 27. The results of the fnc 06 confirm the outperformance of the proposed RTH. All the optimizers provide similar performance for the other two functions with an advantage to the MGO, HGS, and RTH algorithms.

Composition functions (fnc 09–12) ANOVA results are presented in Fig. 28. For function 09, the MGO and the RTH provided the best performance for both cases. For the other functions, the performance of the

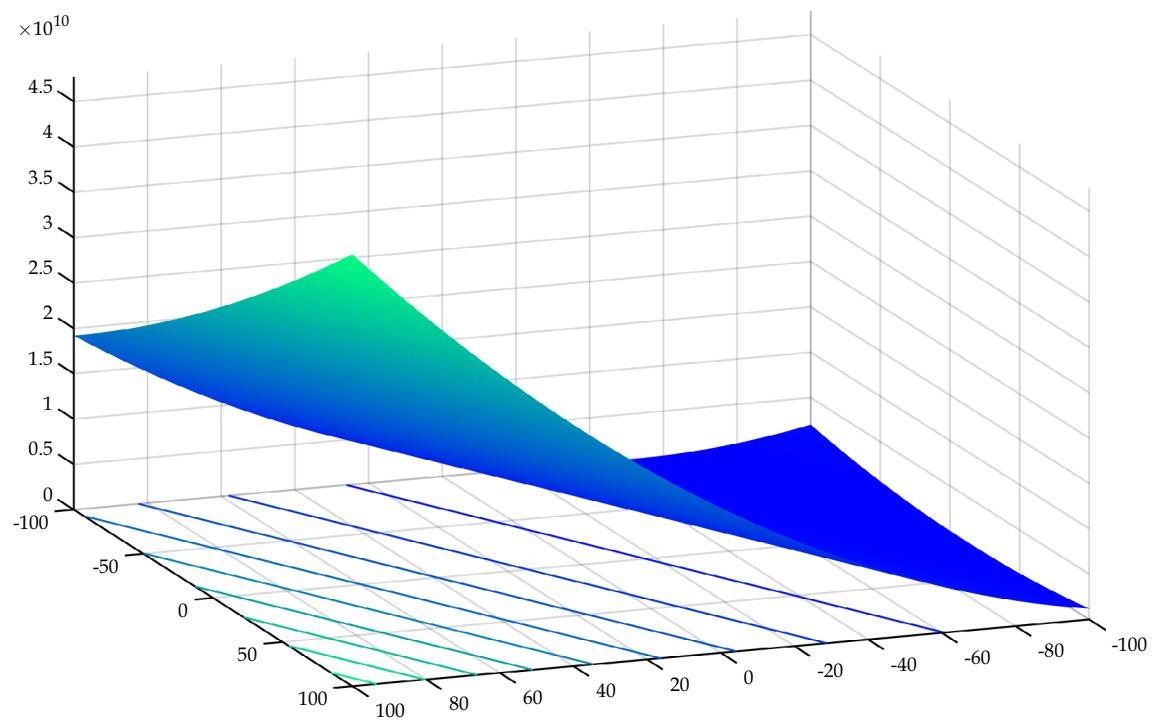


Figure 11. CEC 2020 F1 (unimodal) 3D map.

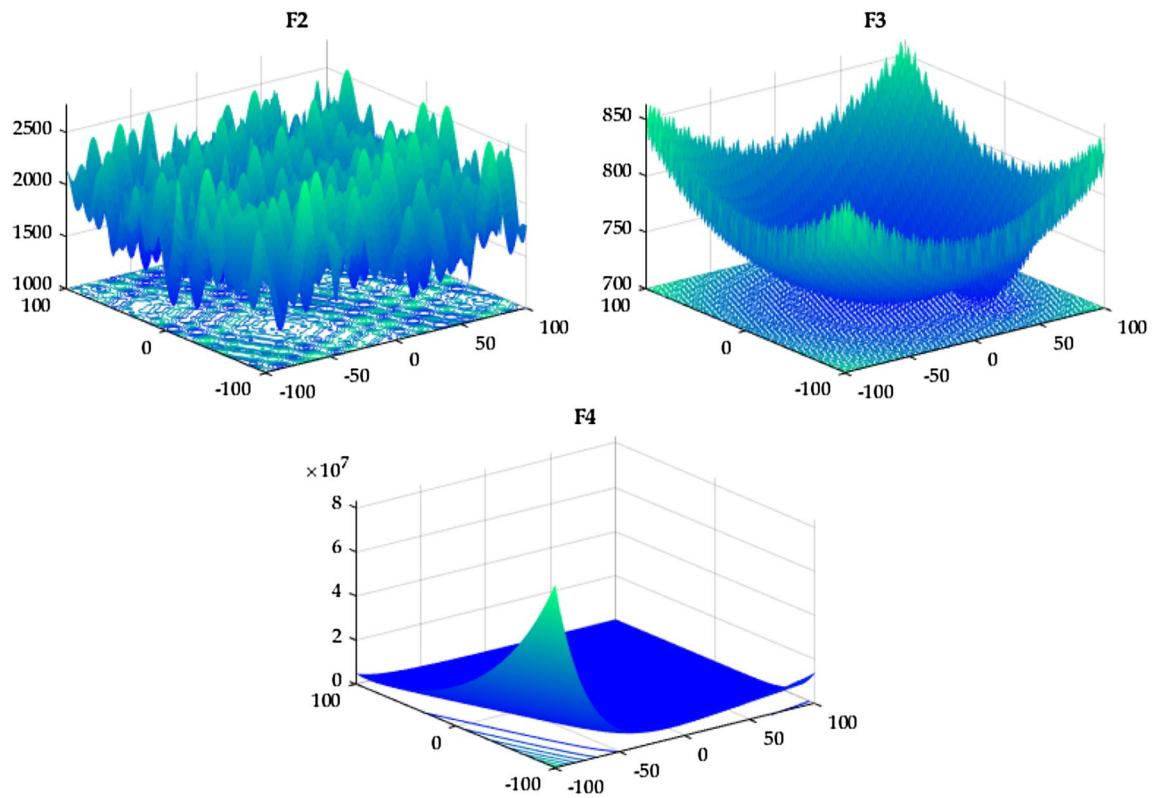


Figure 12. CEC 2020 basic functions' 3D map (F2–F4).

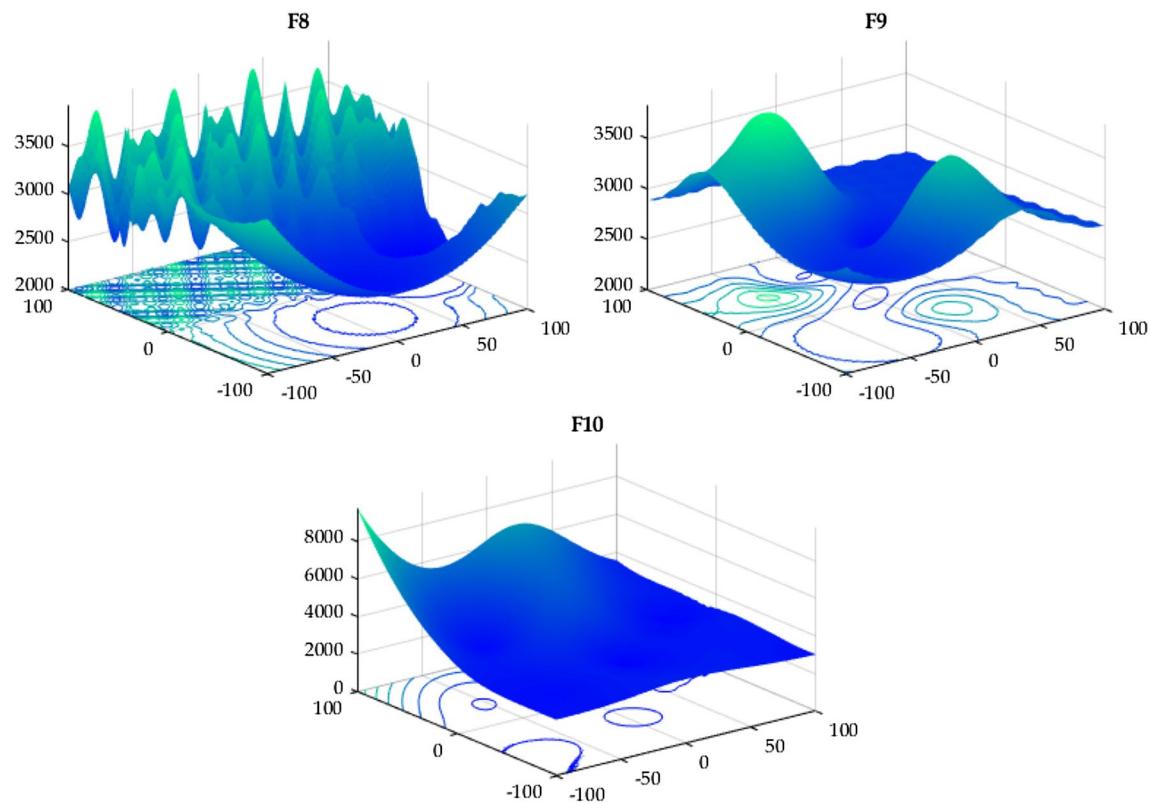


Figure 13. CEC 2020 composition functions' 3D map (F8–F10).

algorithms is much closer to each other, with light superiority for the FO and the HGS for fnc 10, the GTO and the RTH for fnc 11, and the MGO and the COOT for fnc 12.

These mathematical test functions are used to elaborate the performance of the proposed RTH algorithm compared to other new and robust algorithms. The statistical results are summarized in Fig. 29.

From the results obtained, the RTH algorithm can provide very competitive performance for solving various optimization problems. The accuracy and stability of this proposed algorithm have been tested compared to the prementioned algorithms. The convergence speed has been checked and confirmed. After mathematically approving its performance, the proposed RTH algorithm will be tested with some published papers for several engineering tests.

Engineering optimization problems

As mathematics and engineering real-world complex examples, the proposed RTH algorithm will be used to solve seven real-world complex engineering problems.

- (a) *Optimal design of I-Shaped beam:* this issue, which tries to reduce the vertical deflection of the beam, is another common engineering optimization problem³⁴. It satisfies the cross-sectional area and stress restrictions simultaneously under specific loads, as explained in Fig. 30. This optimization problem can be expressed as follows:

$$\begin{aligned}
 f(x) &= \frac{5000}{x_3(x_2 - 2x_4)^3 / 12 + x_1 x_4^3 / 6 + 2bx_4(x_2 - x_4/2)^2} \\
 g_1(x) &= 2x_1 x_3 + x_3(x_2 - 2x_4) \leq 300 \\
 g_2(x) &= \frac{18x_2 \times 10^4}{x_3(x_2 - 2x_4)^3 + 2x_1 x_3(4x_4^2 + 3x_2(x_2 - 2x_4))} + \frac{15x_1 \times 10^3}{(x_2 - 2x_4)x_3^2 + 2x_3 x_1^3} \leq 56 \\
 10 \leq x_1 &\leq 50, \quad 10 \leq x_2 \leq 80, \quad 0.9 \leq x_3 \leq 5, \quad 0.9 \leq x_4 \leq 5
 \end{aligned} \tag{9}$$

where the optimization variables are: the width of the flange (x_1), the height of the section (x_2), the thickness of the web (x_3), and the thickness of the flange (x_4).

From Table 8, the proposed RTH obtains the best solutions for solving this optimization problem compared to other published algorithms. In addition, the PFOA provided similar performance to the proposed RTH, but the RTH achieved the stop criteria after 300 iterations, proving better convergence speed and accuracy.

		MGO	FO	COOT	HGS	AO	HHO	GTO	AVOA	RTH
Unimodal function										
F1	Best	154.539	8576.36×10^6	41887	2181	4024949	4911.13×10^6	104.345	102.058	100.592
	Mean	4974.367	1610.29×10^7	112335	1135972	11527453	1208.31×10^7	5013.844	7186.27	4365.139
	Std $\times 10^3$	4.779	2.867×10^6	45.662	4.50×10^3	8.05×10^3	3.55×10^6	5.93	8.32	4.00
Basic functions										
F2	Best	1126.91	3620.76	1572.07	1412.69	2007.40	2996.31	1678.96	1925.81	1571.65
	Mean	2020.35	3987.02	2334.50	1773.44	2667.60	3857.45	2423.14	2569.96	2401.16
	Std	382.45	184.61	260.14	245.91	387.24	363.03	404.35	294.22	430.07
F3	Best	723.00	911.14	741.18	725.33	774.33	895.80	746.42	766.98	751.74
	Mean	752.73	955.17	768.25	751.06	809.70	963.07	815.69	810.54	806.92
	Std	19.811	19.78	14.66	17.71	21.97	31.88	34.12	32.44	33.36
F4	Best	1901.38	70470.76	1902.476	1901.19	1909.34	1958.76	1902.26	1902.13	1900.08
	Mean	1903.72	294336.46	1906.02	1908.62	1918.19	91339.86	1908.18	1908.71	1903.69
	Std	1	200755.19	1.98	7.30	9.88	82844.70	4.70	3.65	2.66
Hybrid functions										
F5	Best	2492.57	755289	14050	13508.5	96570.2	1270090	1986.35	3909.53	2015.57
	Mean	62543.12	2969214	196629	249879	1098805	11324500	2563.61	244561	2439.74
	Std	57658	1323317	139143	325361	1047277	12488412	286.09	169180	249.82
F6	Best	2319.99								
	Mean	2319.99								
	Std $\times 10^{-3}$	4.47×10^{-8}	3.96×10^{-2}	8.27×10^{-6}	1.39×10^{-9}	6.48×10^{-3}	0.115	1.39×10^{-9}	1.39×10^{-9}	1.39×10^{-9}
F7	Best	2269.802	164541.7	4580.516	5310.281	7155.679	852667.0	2136.214	3331.492	2161.322
	Mean	38399.91	527133.1	66495.89	125533.1	191838.5	6232589	2581.630	94355.00	2597.401
	Std	41656.35	241929.1	88216.10	119168.3	151067.9	3376108	208.5411	108669.8	239.2218
Composition functions										
F8	Best	2300	3107.01	2302.31	2300.51	2308.68	2980.88	2300	2299.52	2300
	Mean	2300.96	3811.25	2304.30	2578.74	2311.52	4326.10	2304.54	2388.05	2556.90
	Std	0.653	381.30	1.44	445.75	0.81	963.94	6.87	338.93	678.38
F9	Best	2796.06	3059.19	2501.81	2500	2550.93	2858.15	2805.19	2801.05	2800
	Mean	2808.72	3133.14	2819.69	2801.70	2846.57	3191.31	2839.22	2836.49	2831.28
	Std	7.79	41.27	62.36	57.56	59.29	138.81	46.35	21.04	15.06
F10	Best	2900	3628.14	2901.42	2900	2909.24	3482.782	2900	2900	2900
	Mean	2935.55	4107.85	3019.66	2973.60	3059.41	4035.529	2975.479	3021.413	2920.14
	Std	84.64	316.40	97.67	100.56	84.6	309.486	102.096	102.84	61.44
L1	Best	3	1	1	2	1	1	2	2	5
L2	Mean	4	1	1	2	1	1	1	1	5
L3	Std	3	1	1	0	0	1	1	1	4

Table 4. CEC 2020 benchmark test function results with D = 15. The best results are marked in bold.

(b) *Three-bar truss design:* In this example, a 3-bar planar truss is considered³⁴, as represented in Fig. 31. The objective function and its constraints are presented as follows

$$\begin{aligned}
 f(x) &= (2\sqrt{2}x_1 + x_2) \times l \\
 g_1(x) &= P \frac{\sqrt{2}x_1 + x_2}{\sqrt{2x_1^2 + 2x_2x_1}} - \sigma \leq 0 \\
 g_2(x) &= P \frac{x_2}{\sqrt{2x_1^2 + 2x_2x_1}} - \sigma \leq 0 \\
 g_3(x) &= P \frac{x_2}{\sqrt{2x_2 + x_1}} - \sigma \leq 0 \\
 l &= 100\text{cm}, \quad P = 2\text{kN/cm}^3, \quad \sigma = 2\text{kN/cm}^3, \quad 0 \leq x_1, \quad x_2 \leq 1
 \end{aligned} \tag{10}$$

Table 9 provides the statistical results of these algorithms. It is clear that the best results are obtained using the proposed RTH algorithm and the AHA optimizer. However, the required number of iterations is only 670,

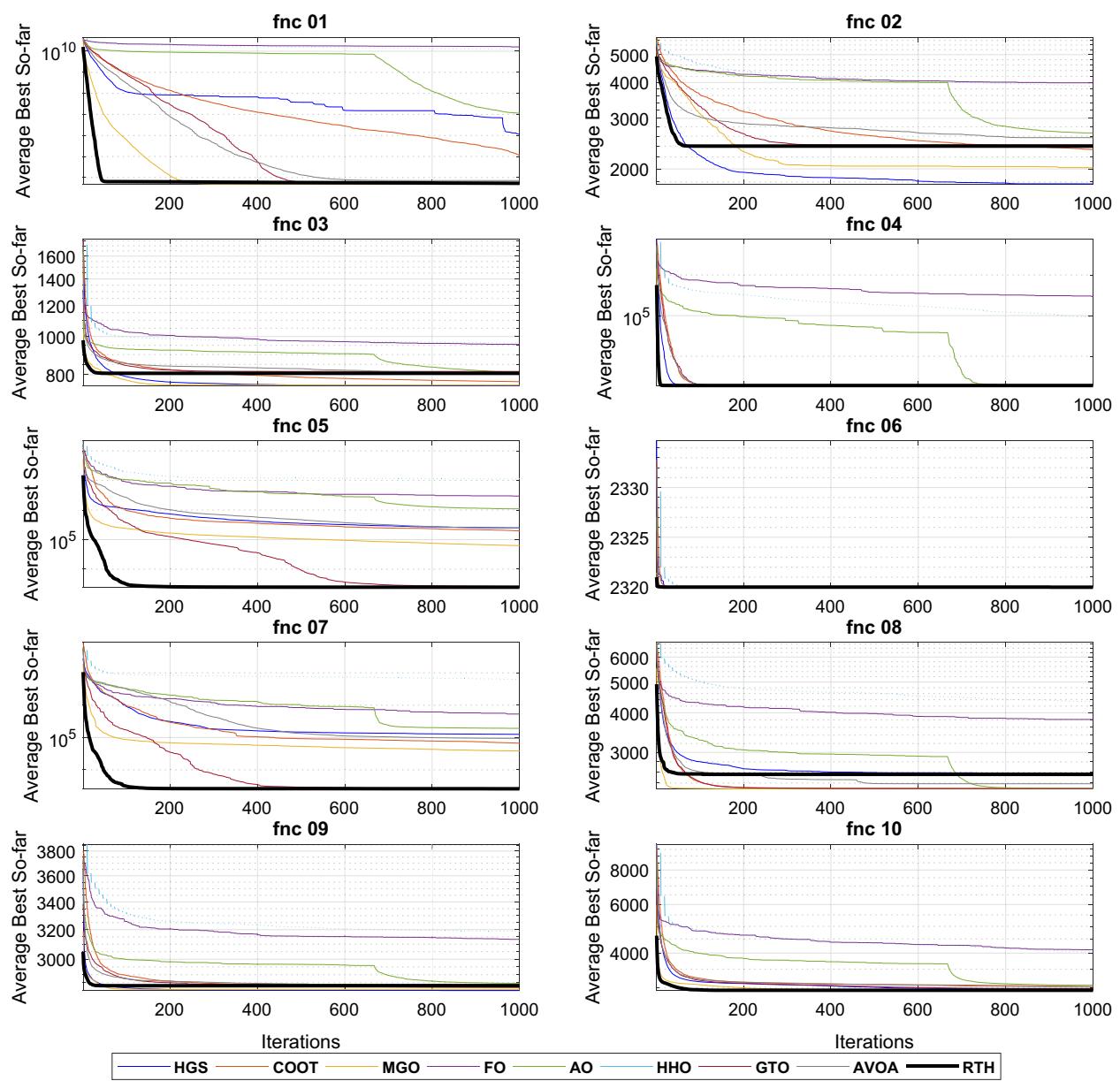


Figure 14. CEC 2020 fitness evolution for $D = 15$.

much lower than the other reported algorithms, including the AHA, which requires 15,000 iterations to achieve the optimal value.

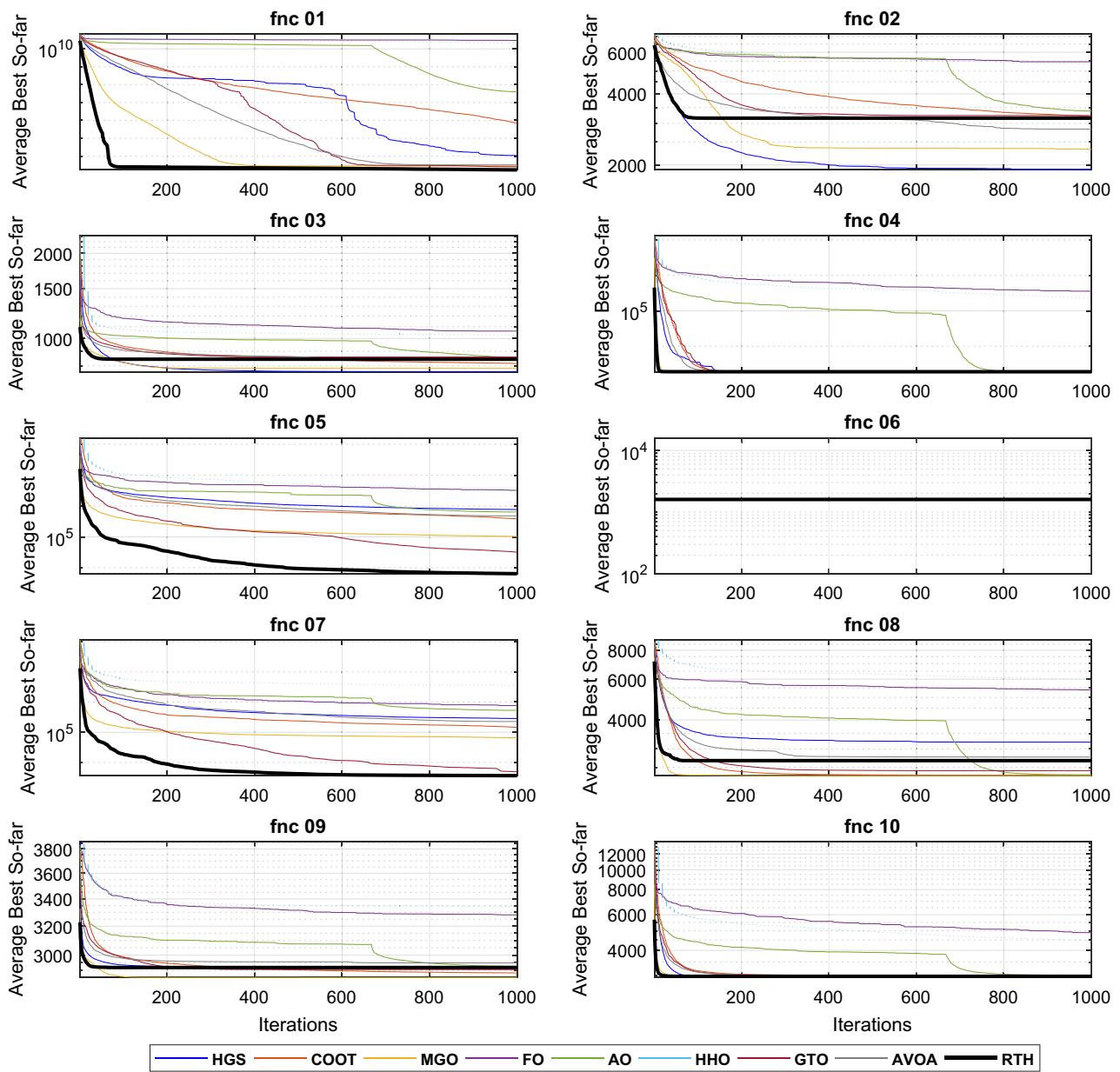
- (c) *Design of a tubular column:* this problem illustrates the structure of a uniform column with a tubular cross-section that can support a compressive load at the lowest possible cost³⁴, as explained in Fig. 32. The two design variables in this issue are the mean diameter of the column (x_1) and the thickness of the tube (x_2). The column is constructed from a material having a yield stress of y and an elasticity modulus of E . The following equation represents this optimization problem:

		MGO	FO	COOT	HGS	AO	HHO	GTO	AVOA	RTH
Unimodal function										
F1	Best	103.05	16806140527	224340.05	130.33	7001885.6	13513137381	100.25	100.49	100.20
	Mean	2840.68	29506906789	637140.05	10976.98	39584617.2	21692845681	2554.76	3266.99	1715.68
	StD $\times 10^3$	3312.12	4605945120	311729.08	16272.32	21104674.4	4650504699	2610.12	3863.87	1565.21
Basic functions										
F2	Best	1472.78	4746.85	2112.23	1455.18	2653.59	4630.30	2276.34	1926.65	1966.20
	Mean	2333.59	5471.61	3206.10	1905.89	3385.10	5383.45	3238.12	2838.21	3156.77
	StD	438.85	259.64	452.76	220.67	474.10	357.14	463.24	482.15	596.00
F3	Best	743.17	982.24	776.23	735.22	803.50	931.26	798.78	785.60	796.21
	Mean	783.98	1061.86	815.46	762.18	855.62	1028.41	859.93	836.36	844.61
	StD	20.88	36.71	28.20	17.65	27.78	31.34	41.12	30.18	30.29
F4	Best	1902.61	98274.78	1905.98	1901.75	1915.55	9210.16	1905.96	1906.25	1900.93
	Mean	1910	358004.03	1910.40	1903.68	1932.14	227855.65	1918.30	1914.45	1909.82
	StD	9.30	246927.96	2.57	1.20	12.14	190896.04	9.38	6.26	4.61
Hybrid functions										
F5	Best	13793.98	963554.04	71241.37	35642.41	180500.1	258875.09	4762.99	25666.73	2793.50
	Mean	103507.19	3221412.17	384815.44	769776.7	640537.1	3585480.15	32023.92	480253.8	6392.68
	StD	60422.30	1429844.08	239613.83	527020.1	339804.200	2798617.66	34478.79	413365.6	3120.73
F6	Best	1604.05								
	Mean	1604.05								
	StD $\times 10^{-12}$	1.16								
F7	Best	3544.55	271391.75	9867.50	34895.32	42748.08	318315.91	2621.89	19735.31	2443.60
	Mean	65648.76	763153.83	144945.05	284910.30	530727.72	3573597.72	4921.82	226243.15	3648.84
	StD	57276.90	423466.26	92024.16	253278.10	450075.13	3473899.96	1940.52	189687.04	947.33
Composition functions										
F8	Best	2300	4347.34	2306.97	2300	2314.95	3880.60	2300	2300	2300
	Mean	2300.70	5396.11	2309.54	3209.51	2321.03	6004.87	2418.29	2768.84	2671.51
	StD	0.63	576.35	1.13	1247.79	4.81	892.123	639.17	1107.05	1004.51
F9	Best	2821.03	3125.32	2821.07	2850.66	2863.87	3081.44	2837.50	2860.88	2861.11
	Mean	2853.78	3281.45	2883.40	2917.45	2927.31	3341.05	2905.20	2948.24	2918.42
	StD	26.25	59.29	37.24	34.10	44.59	149.48	49.74	49.63	40.15
F10	Best	2913.93	4160.30	2911.42	2909.25	2941.17	3605.15	2913.10	2910.51	2901.27
	Mean	2973.47	4890.83	2979.09	2931.18	3010.99	4473.18	2969.76	2982.71	2959.72
	StD	33.02	539.25	31.04	28.29	23.33	643.61	35.28	34.40	31.12
L1	Best	3	1	1	2	1	1	2	2	6
L2	Mean	3	1	1	2	1	1	1	1	5
L3	StD	2	1	1	2	1	1	1	1	5

Table 5. CEC 2020 benchmark test function results with D = 15. The best results are marked in bold.

$$\begin{aligned}
 f(x) &= 9.8x_1x_2 + 2x_2 \\
 g_1(x) &= \frac{P}{\pi x_2 x_1 \sigma_y} - 1 \leq 0 \\
 g_2(x) &= \frac{8PL^2}{\pi^3 E x_2 x_1 (x_1^2 + x_2^2)} - 1 \leq 0 \\
 g_3(x) &= \frac{2}{x_1} - 1 \leq 0, \quad g_4(x) = \frac{x_1}{14} - 1 \leq 0, \\
 g_5(x) &= \frac{2}{x_2} - 1 \leq 0, \quad g_6(x) = \frac{x_2}{8} - 1 \leq 0 \\
 \sigma_y &= 500 \text{kgf/cm}^2, \quad P = 0.85 \times 10^6 \text{kgf/cm}^3, \quad 2 \leq x_1 \leq 14, \quad 0.2 \leq x_2 \leq 0.8
 \end{aligned} \tag{11}$$

This problem has already been tackled using several algorithms. The best results of these algorithms, as well as the suggested RTH, are shown in Table 10. Based on these results, the proposed RTH achieved the best results with the lowest number of iterations (300).

**Figure 15.** CEC 2020 fitness evolution for $D = 20$.

- (d) *Speed Reducer Design:* speed reducer is an essential part of the gearbox in mechanical systems. It may be used for a variety of purposes⁷¹. As shown in Fig. 33, the weight of the speed reducer must be reduced under 11 restrictions in this optimization problem. Face width (x_1), the module of teeth (x_2), number of teeth in the pinion (x_3), length of the first shaft between bearings (x_4), length of the second shaft between bearings (x_5), the diameter of first shafts (x_6), and diameter of the second shafts (x_7) are the seven variables in this issue. This problem can be modeled as

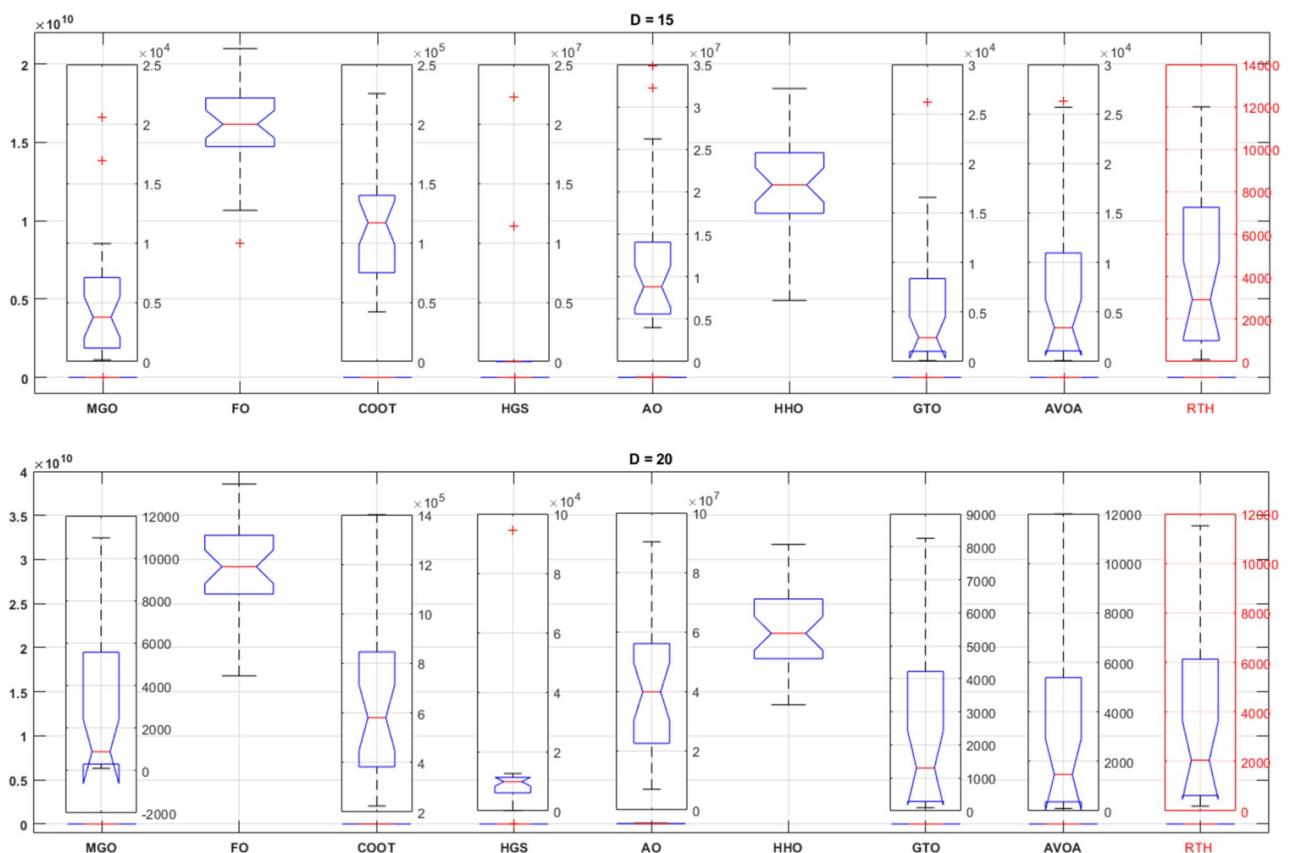


Figure 16. CEC 2020 unimodal function (F1) ANOVA results.

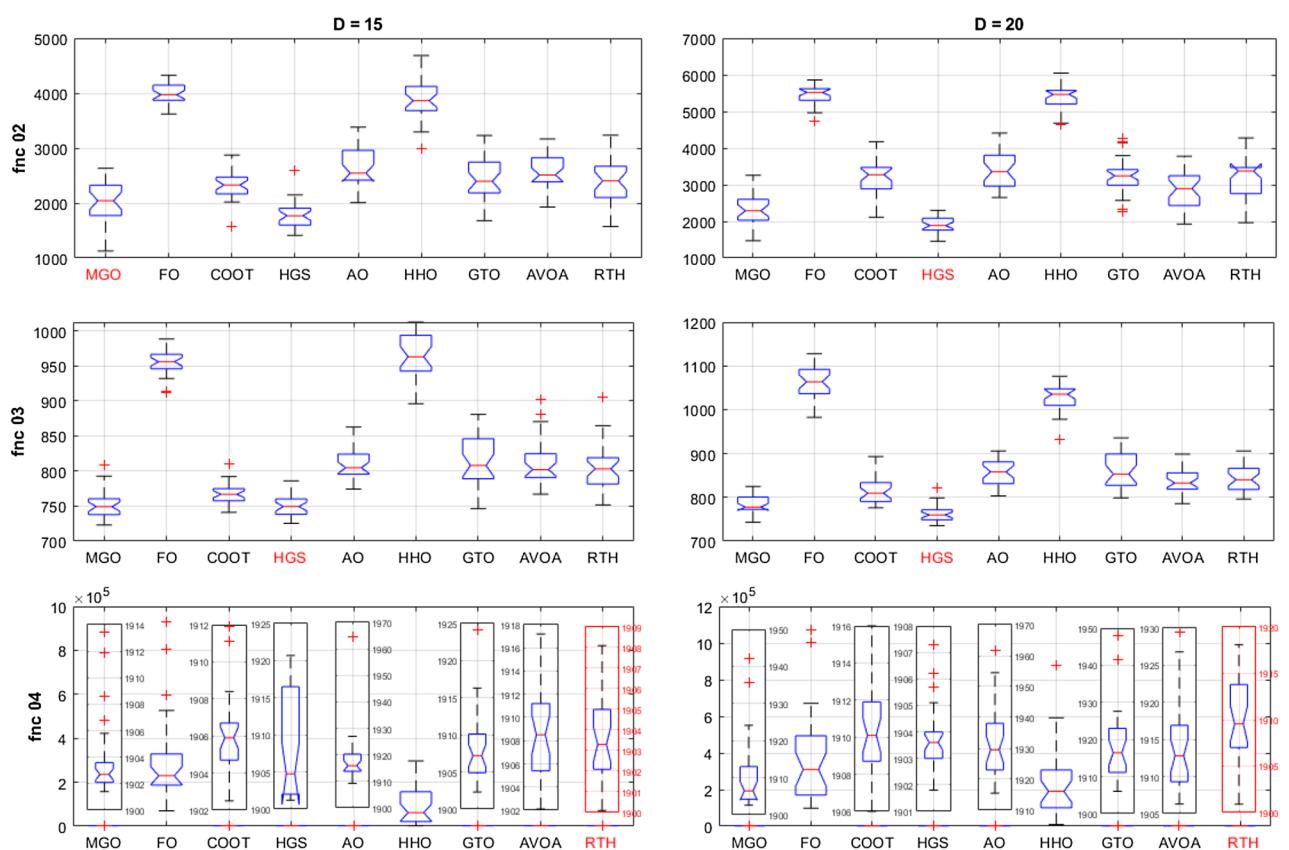


Figure 17. CEC 2020 basic functions ANOVA results.

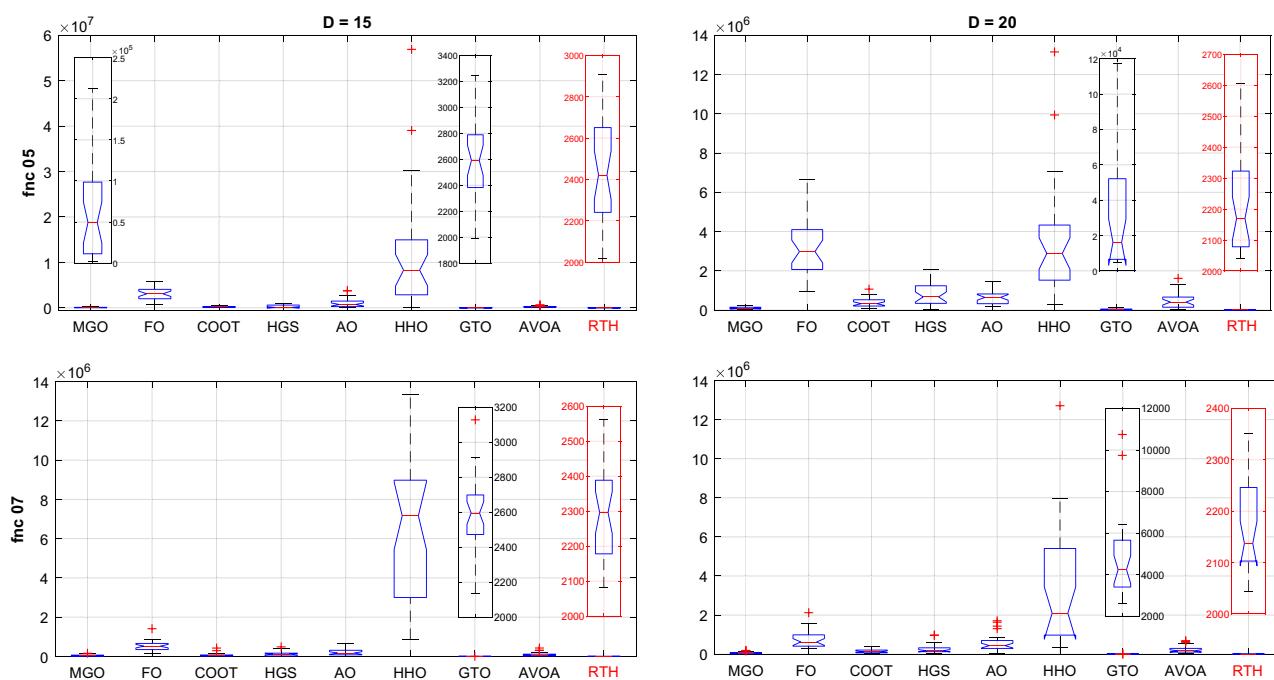


Figure 18. CEC 2020 hybrid functions ANOVA results.

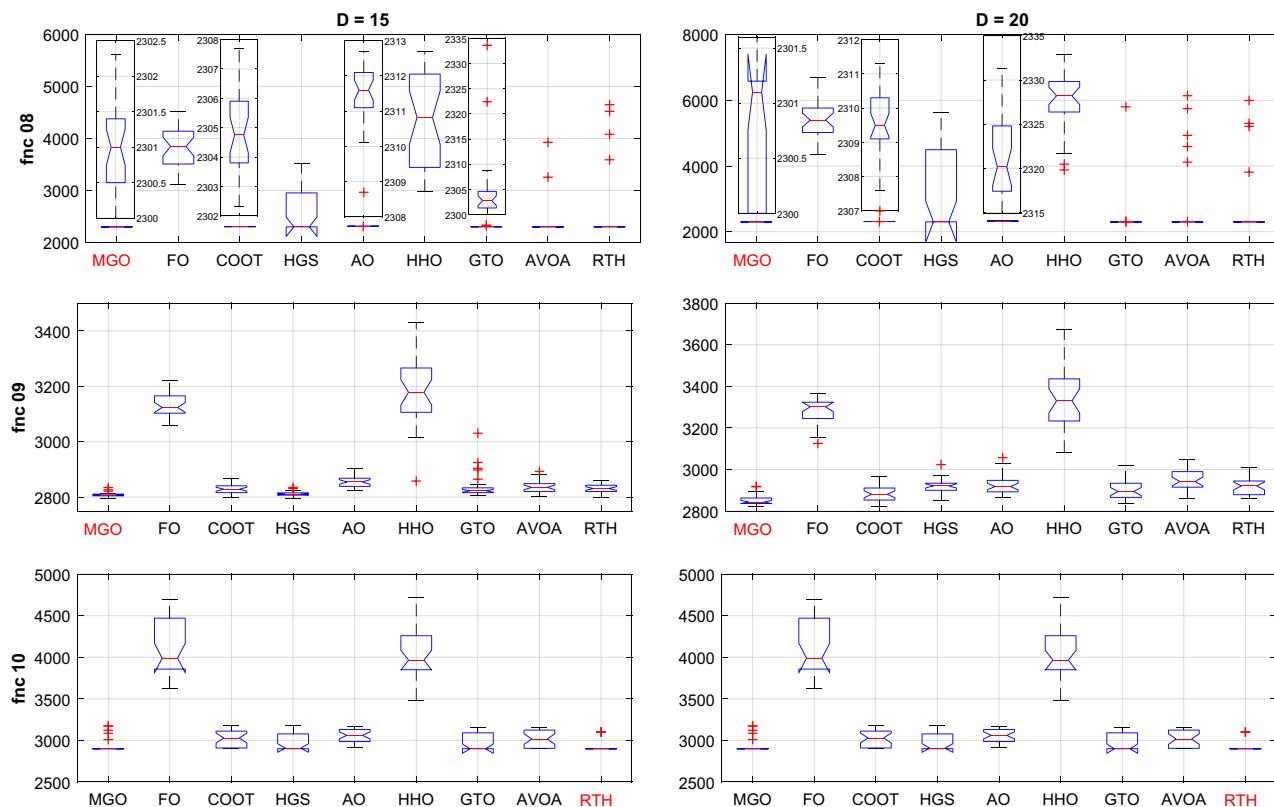


Figure 19. CEC 2020 composite functions ANOVA results.

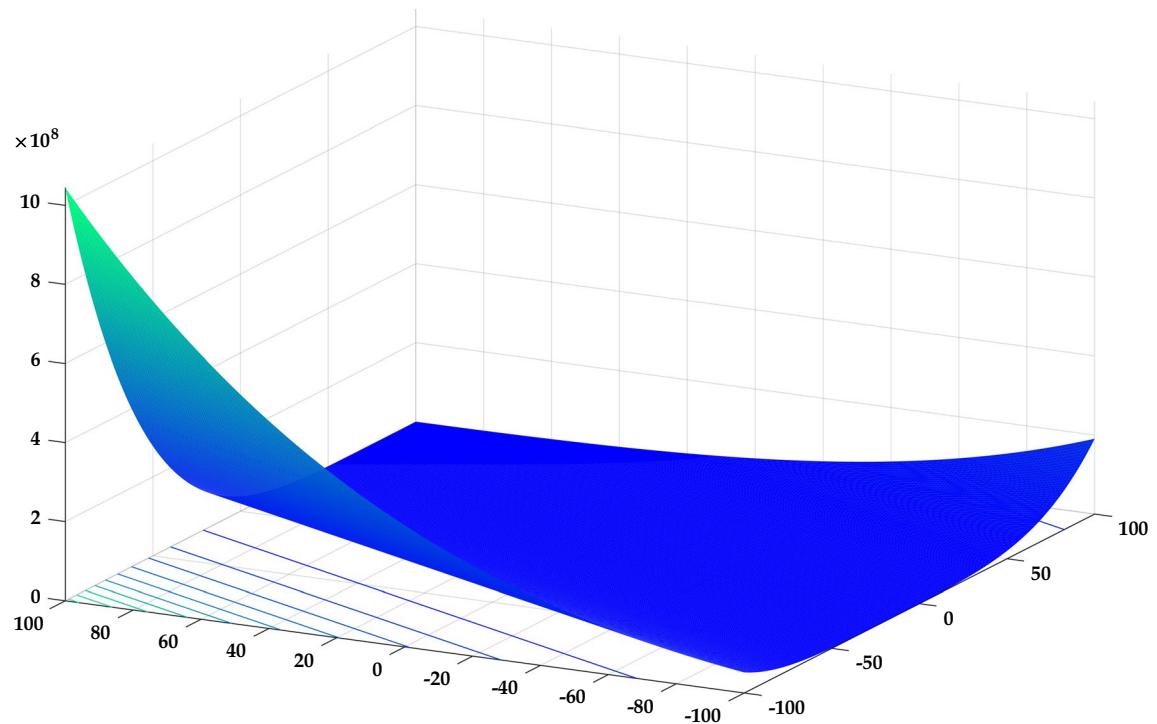


Figure 20. CEC 2022 F1 (unimodal) 3D map.

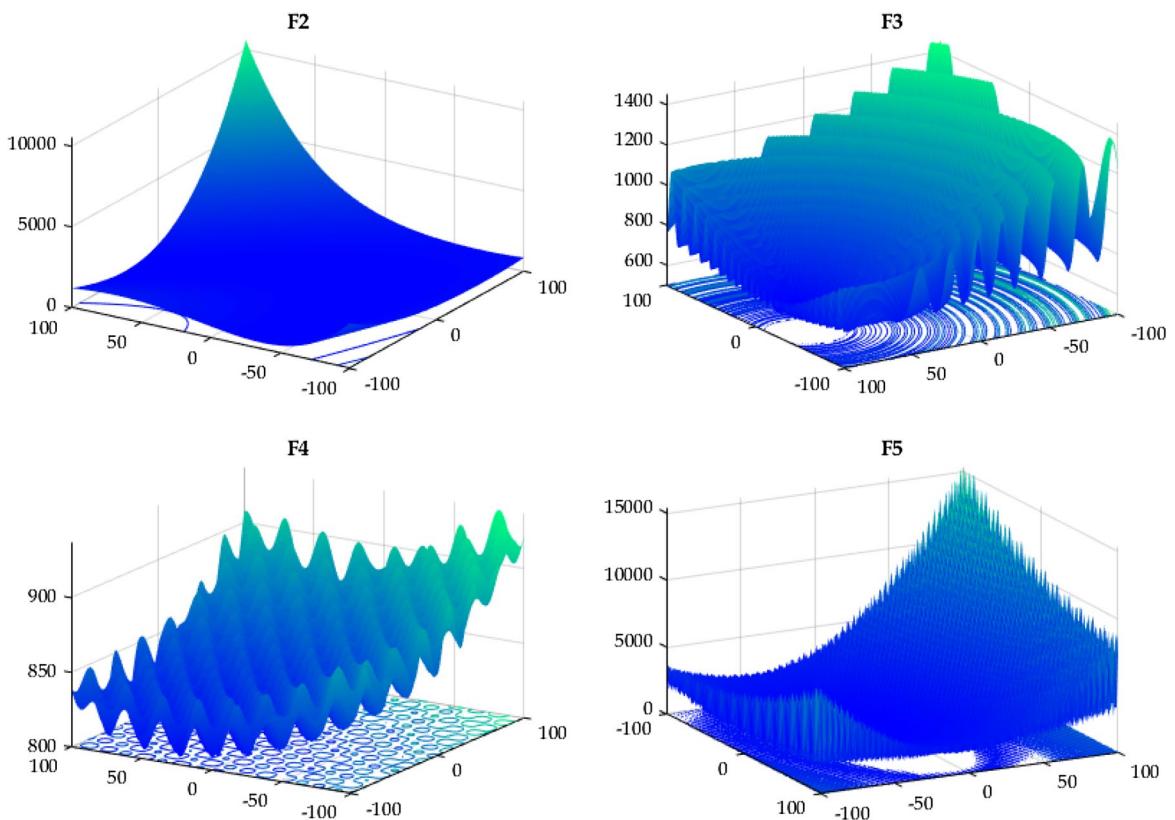


Figure 21. CEC 2022 basic functions' 3D map (F2–F5).

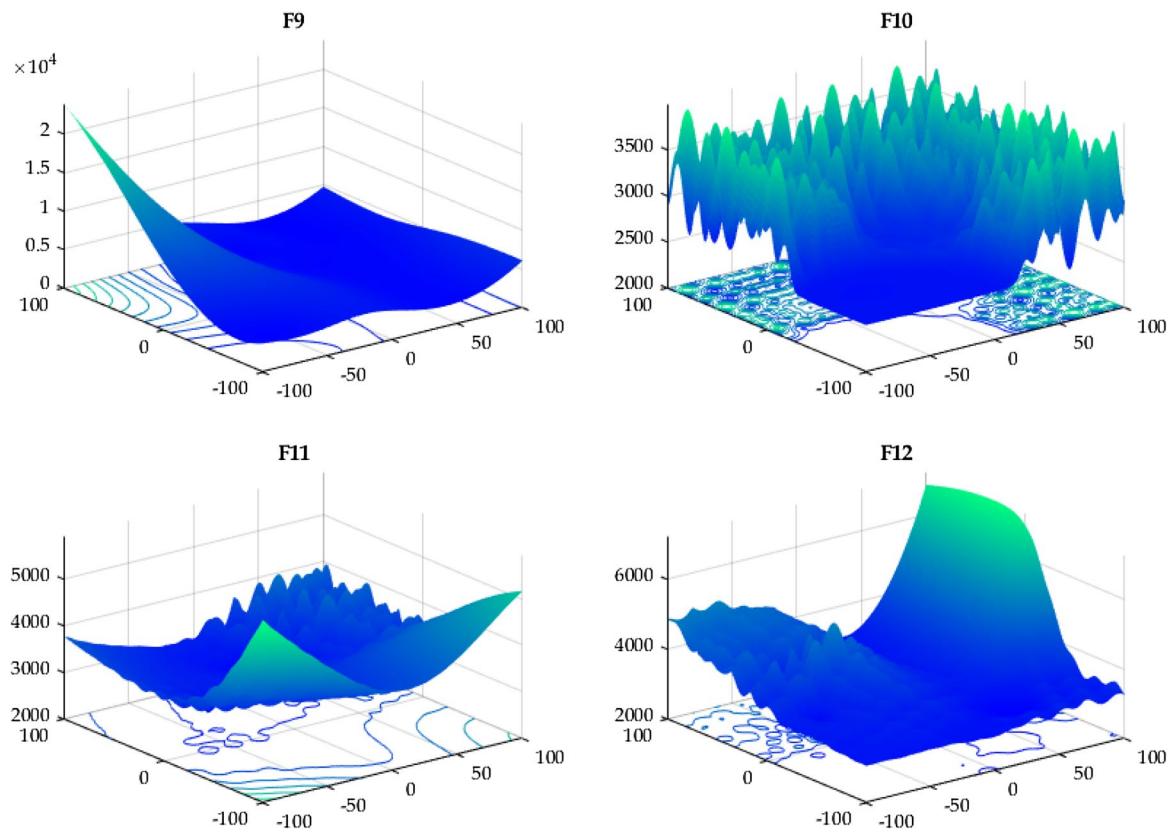


Figure 22. CEC 2022 composition functions' 3D map (F9–F12).

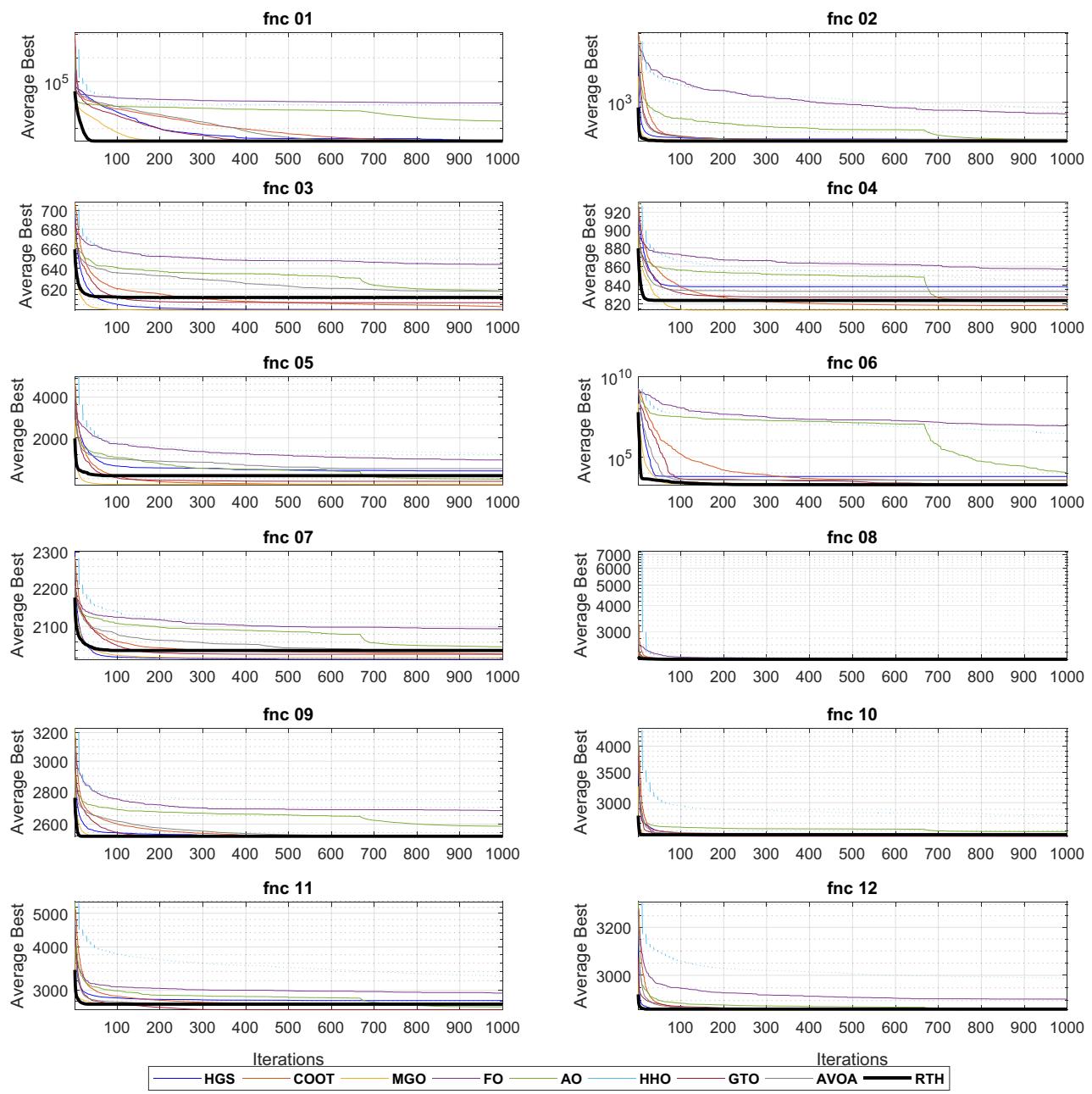
$$\begin{aligned}
 f(x) &= 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) \\
 &\quad - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2) \\
 g_1(x) &= \frac{27}{x_1x_3x_2^2} - 1 \leq 0, \quad g_2(x) = \frac{397.5}{x_1x_3x_2^2} - 1 \leq 0 \\
 g_3(x) &= \frac{1.93}{x_2x_3x_6^4} - 1 \leq 0, \quad g_4(x) = \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \leq 0 \\
 g_5(x) &= \frac{\sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2 + 16.9 \times 10^6}}{110x_6^3} - 1 \leq 0 \\
 g_6(x) &= \frac{\sqrt{\left(\frac{745x_5}{x_2x_3}\right)^2 + 157.5 \times 10^6}}{85x_7^3} - 1 \leq 0 \\
 g_7(x) &= \frac{x_2x_3}{40} - 1 \leq 0, \quad g_8(x) = \frac{5x_2}{x_1} - 1 \leq 0, \quad g_9(x) = \frac{x_1}{12x_2} - 1 \leq 0 \\
 g_{10}(x) &= \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0, \quad g_9(x) = \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0 \\
 2.6 \leq x_1 &\leq 3.6, \quad 0.7 \leq x_2 \leq 0.8, \quad x_3 \in [17, 18, \dots, 28], \quad x_4 \geq 7.3 \\
 x_5 \leq 8.3, \quad 2.9 \leq x_6 &\leq 3.9, \quad 5 \leq x_7 \leq 5.5
 \end{aligned} \tag{12}$$

The results of RTH and other published algorithms are reported in Table 11. Among these algorithms, the RTH has the lowest number of iterations, while its metrics in terms of best, mean, and StD are better than those other algorithms.

- (e) *Piston lever:* the essential goal of this problem is to identify the piston components (x_1), (x_2), (x_3), and (x_4) by reducing the oil volume when the piston's lever is raised from 0° to 45° ³⁴. This problem is presented in Fig. 34, and it can be modeled as follows:

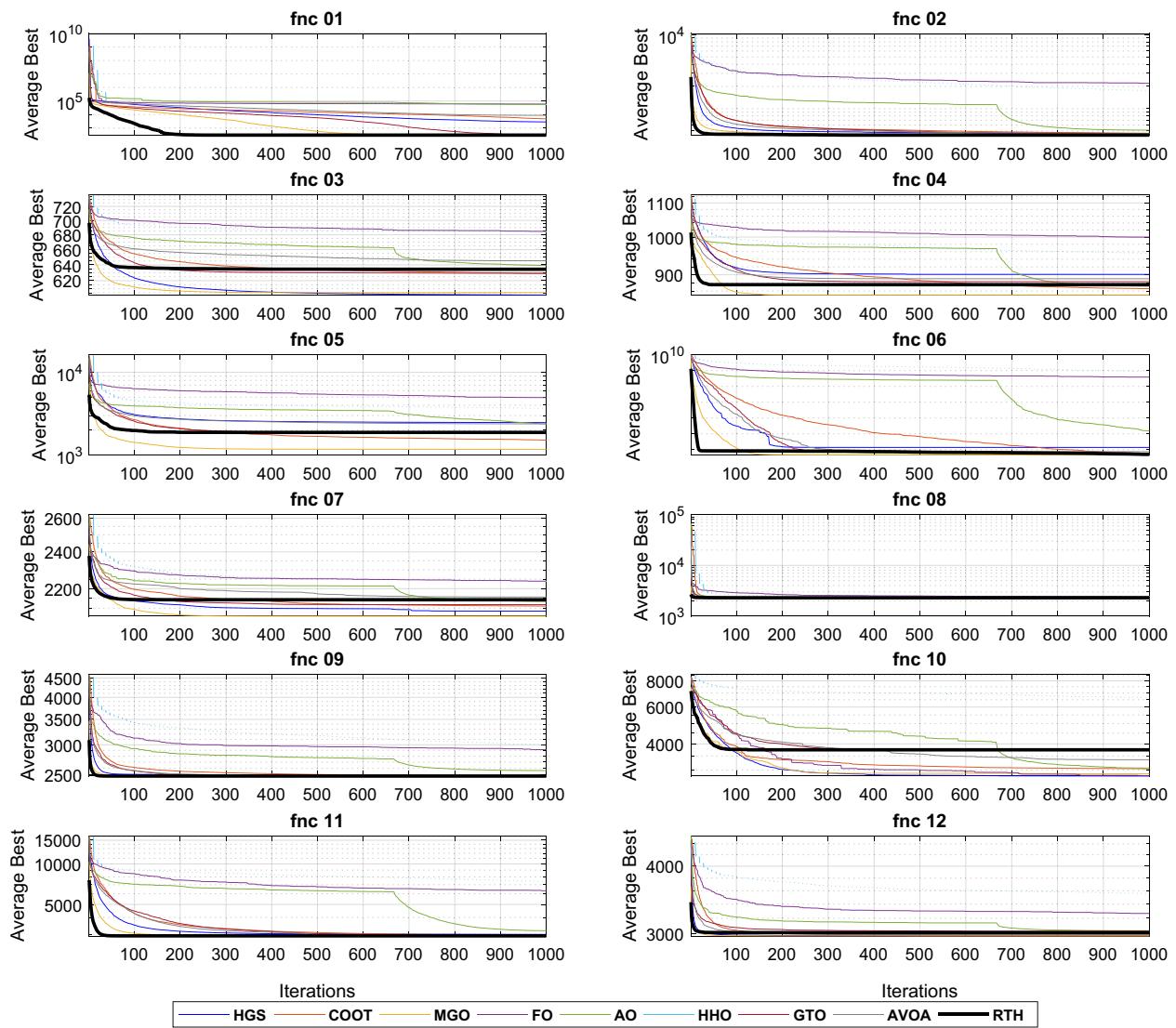
	MGO	FO	COOT	HGS	AO	HHO	GTO	AVOA	RTH
Unimodal function									
F1	Best	300	7075.29	300.23	300	439.34	5774.01	300	300
	Mean	300	12242.75	302.53	321.05	2127.41	9833.45	300	302.72
	StD	4.09×10^{-12}	3256.05	2.71	115.26	1730.22	1056.31	1.93E-10	8.05
Basic functions									
F2	Best	400.001	519.53	400.002	400.39	400.189	502.241	400.002	400.013
	Mean	405.32	765.59	419.209	421.58	421.727	850.32	408.968	415.57
	StD	3.77	153.57	28.72	29.89	36.835	321.15	17.82	25.613
F3	Best	600	633.70	600.40	600	605.12	626.52	601.18	602.62
	Mean	600.17	643.93	603.26	600.67	618.32	648.29	607.01	617.85
	StD	0.25	5.38	2.44	0.60	6.92	11.60	5.17	10.68
F4	Best	802.99	839.46	807.34	813.93	810.07	831.72	808.96	808.96
	Mean	813.73	856.94	818.33	838.07	823.00	844.06	827.20	833.13
	StD	7.25	6.92	5.92	14.68	8.21	9.92	7.41	11.01
F5	Best	900	1170.61	900.01	900.91	903.56	1142.71	902.40	955.37
	Mean	904.88	1375.01	902.57	1141.03	993.59	1537.75	956.74	1186.98
	StD	9.07	124.92	4.16	252.20	81.42	205.56	66.27	196.57
Hybrid functions									
F6	Best	1846.46	430643.97	1917.56	1835.65	3197.23	3007.22	1805.15	1877.29
	Mean	2122.37	8578353.07	3444.352	6035.25	10648.95	2588374.51	2008.43	3592.39
	StD	344.20	7161565.78	1736.32	2355.11	5945.94	8077679.39	707.50	1969.34
F7	Best	2001	2066.31	2009.90	2000.01	2021.64	2039.65	2006.52	2020.59
	Mean	2020.96	2094	2029.99	2016.77	2048.19	2106.54	2029.45	2038.85
	StD	6.29	11.28	9.91	7.44	24.12	30.63	11.62	20.07
F8	Best	2214.70	2233.23	2208.56	2219.43	2208.16	2228.73	2206.81	2221.88
	Mean	2226.69	2246.53	2224.47	2222.47	2227.78	2249.43	2221.81	2225.57
	StD	21.87	6.53	4.02	3.47	6.11	24.51	3.39	3.35
Composition functions									
F9	Best	2529.28	2635.33	2529.28	2529.28	2530.96	2634.54	2529.28	2529.28
	Mean	2529.28	2680.84	2529.29	2531.32	2587.69	2730.09	2529.45	2534.18
	StD	0	21.64	0.03	11.14	32.55	45.59	0.59	26.83
F10	Best	2500.22	2509.88	2500.33	2500.41	2500.71	2502.28	2500.31	2500.37
	Mean	2531.19	2528.46	2546.63	2557.65	2588.60	2812.47	2527.34	2564.39
	StD	52.07	10.33	57.59	61.64	54.02	455.78	49.60	65.20
F11	Best	2600	2822.82	2600.39	2600	2604.60	2787.33	2600	2600
	Mean	2721.97	2946.83	2731.62	2805.30	2690.65	3284.55	2638.41	2719.58
	StD	123.03	81.79	183.31	170.69	93.38	406.47	88.85	138.41
F12	Best	2861.41	2884.28	2859.00	2861.41	2862.75	2873.20	2861.44	2861.41
	Mean	2864.73	2905.10	2864.85	2864.07	2867.99	2985.81	2864.64	2866.86
	StD	1.49	13.50	2.33	1.43	3.45	78.94	2.13	5.65
L1	6	0	1	5	0	0	3	4	8
L2	6	0	0	2	0	0	2	0	5
L3	4	2	0	1	0	0	0	0	5

Table 6. CEC 2022 benchmark test function results for D = 10. The best results are marked in bold.

**Figure 23.** CEC 2022 fitness evolution for D=10.

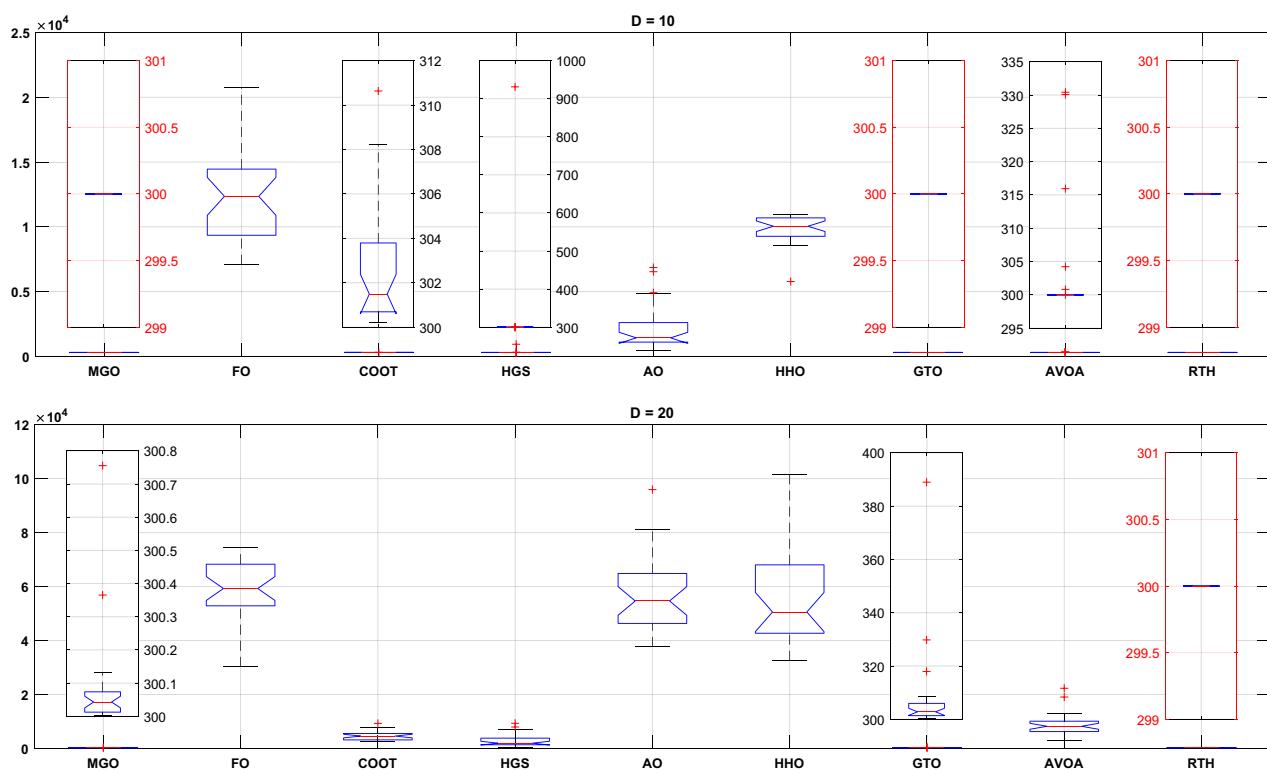
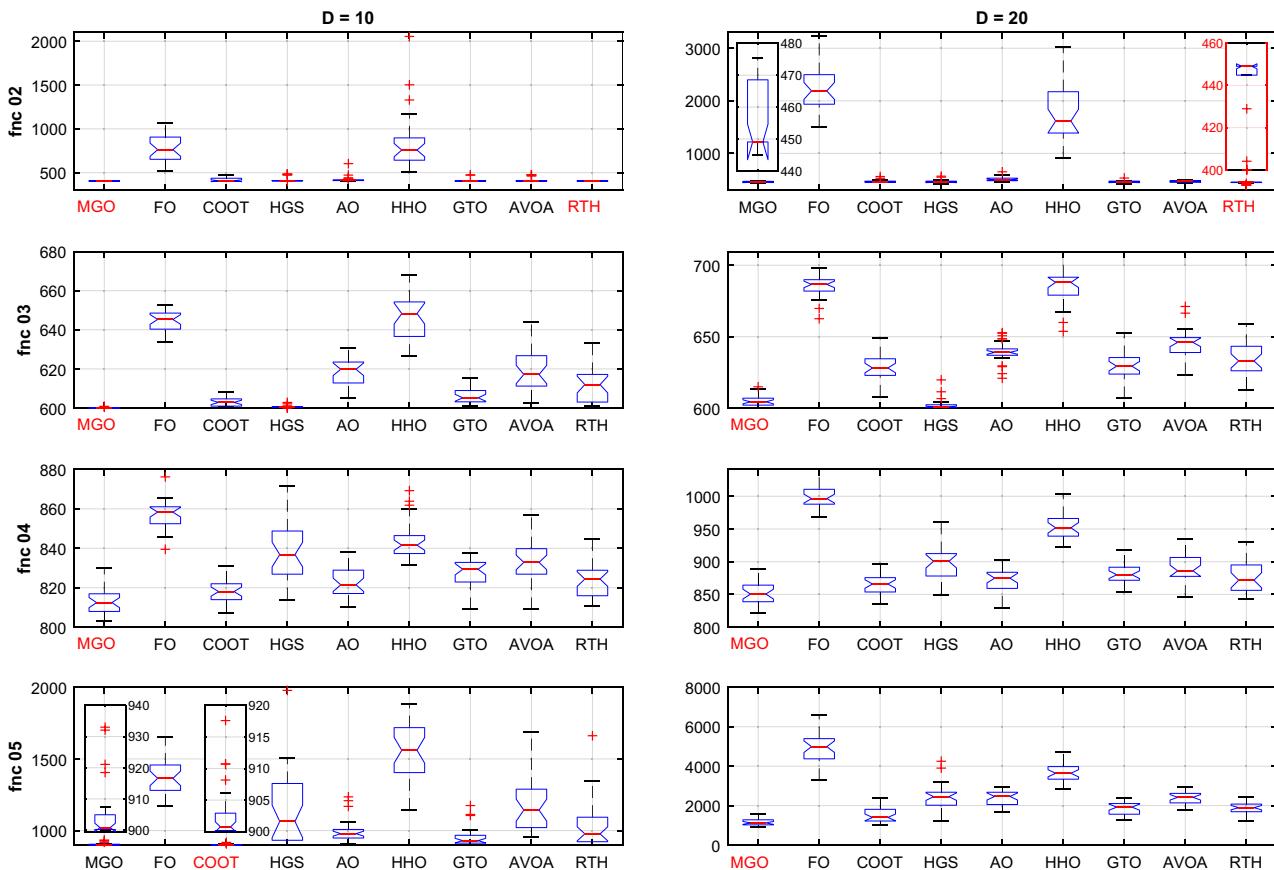
	MGO	FO	COOT	HGS	AO	HHO	GTO	AVOA	RTH	
Unimodal function										
F1	Best	300.02	30,224.81	2435.79	477.80	37,670.33	32,487.71	300.16	3008.42	300
	Mean	300.08	59,629.62	4778.73	2836.77	56,386.78	56,267.25	307.43	8776.13	300
	StD	0.145	9986.83	1631.82	2289.21	13,929.99	20,389.83	16.521	4062.34	9.5×10^{-12}
Basic functions										
F2	Best	444.90	1504.50	449.28	415.00	463.19	912.12	414.82	428.68	400
	Mean	454.95	2200.73	465.98	462.83	513.93	1750.55	458.46	464.59	439.81
	StD	10.311	435.219	24.26	30.513	44.726	529.31	19.660	15.015	18.155
F3	Best	600.24	662.56	607.77	600.14	620.86	653.72	606.71	623.472	612.84
	Mean	605.31	684.74	629.01	602.34	639.25	684.30	629.32	645.34	634.42
	StD	4.190	7.393	9.322	4.048	7.358	12.220	10.058	9.661	11.915
F4	Best	821.89	968.55	835.56	849.58	829.52	922.31	853.73	845.77	842.78
	Mean	850.46	999.82	865.50	900.93	871.90	954.79	882.85	890.06	875.42
	StD	16.05	17.376	15.214	29.79	16.95	20.278	17.669	23.452	22.258
F5	Best	930.94	3277.01	992.25	1238.17	1666.064	2849.926	1249.81	1788.12	1206.33
	Mean	1169.31	4948.35	1513.81	2462.87	2371.00	3673.03	1848.52	2378.20	1874.44
	StD	194.436	859.562	376.653	636.69	375.57	471.658	343.371	304.611	324.049
Hybrid functions										
F6	Best	1882.54	90,365,085.95	2223.48	1947.71	25,861.92	57,613,430.85	1862.02	2022.830	1800.30
	Mean	4612.37	389,128,839.2	5328.58	13,826.44	155,894.16	867,615,884.2	5794.91	7461.50	4610.99
	StD	3201.410	190,973,416.5	3227.276	9392.515	117,944.76	838,420,299.2	4755.08	5751.65	2683.249
F7	Best	2027.05	2192.06	2052.78	2023.81	2070.94	2172.70	2035.89	2027.46	2069.44
	Mean	2060.66	2239.15	2110.38	2085.49	2133.50	2239.80	2117.42	2156.65	2142.50
	StD	26.909	26.480	38.913	52.669	72.288	46.489	40.241	66.119	55.245
F8	Best	2220.77	2286.92	2228.55	2221.19	2229.99	2238.83	2221.56	2228.03	2210.52
	Mean	2224.82	2418.70	2267.21	2253.21	2247.24	2422.96	2243.35	2238.87	2246.50
	StD	6.695	70.141	54.040	47.844	31.615	148.88	43.417	9.448	26.753
Composition functions										
F9	Best	2480.782	2720.715	2480.884	2480.782	2494.789	2639.303	2480.781	2480.782	2480.781
	Mean	2480.797	2913.051	2485.666	2486.190	2565.406	3000.960	2480.814	2483.454	2480.781
	StD	0.013	90.519	5.330	5.828	43.243	286.987	0.048	3.432	3.63×10^{-12}
F10	Best	2500.472	2610.636	2500.668	2414.572	2500.670	5714.266	2500.754	2500.760	2500.745
	Mean	2878.745	2820.891	3040.516	2806.015	3065.910	6675.800	3726.354	3359.881	3745.876
	StD	639.116	160.398	855.475	165.485	910.325	514.280	1165.580	767.330	726.295
F11	Best	2900	5193.046	2619.646	2600.055	2876.708	5654.721	2900	2600.002	2900
	Mean	2930	6317.223	2941.369	2990.848	3186.476	7419.084	2965.112	2910.003	2916.667
	StD	46.609	539.848	167.852	161.029	185.885	816.256	88.733	95.952	37.905
F12	Best	2939.506	3158.671	2943.284	2940.273	2983.960	3213.355	2946.685	2940.228	2938.921
	Mean	2984.007	3266.632	2973.659	2967.016	3033.970	3559.546	3026.821	2981.706	2972.224
	StD	57.231	55.531	24.066	29.855	40.245	254.008	59.346	47.495	30.449
L1		3	0	0	2	0	0	0	1	6
L2		5	0	0	1	0	0	0	1	4
L3		3	2	1	1	0	0	0	0	5

Table 7. CEC 2022 benchmark test function results for D = 10. The best results are marked in bold.

**Figure 24.** CEC 2022 fitness evolution for $D = 10$.

$$\begin{aligned}
 f(x) &= \frac{1}{4}\pi x_3^2(L_2 - L_1) \\
 g_1(x) &= QL \cos(\theta) - RF \leq 0 \\
 g_2(x) &= Q(L - x_4) - M_{\max} \leq 0 \\
 g_3(x) &= 1.2(L_2 - L_1) - L_1 \leq 0 \\
 g_4(x) &= \frac{x_3}{2} - x_2 \leq 0 \\
 R &= \frac{-x_4(x_4 \sin(\theta) + x_1) + x_1(x_2 - x_4 \cos(\theta))}{L_1} \\
 F &= \pi Px_3^2/4, \quad L_1 = \sqrt{(x_4 - x_2)^2 + x_1^2}, \quad L_2 = \sqrt{(x_4 \sin(\theta) + x_1)^2 + (x_2 - x_4 \cos(\theta))^2} \\
 \theta &= 45^\circ, \quad Q = 10000 \text{lbs}, \quad L = 240 \text{in}, \quad 1.8 \times 10^6 \text{lbs.in}, \quad P = 1500 \text{psi} \\
 0.05 \leq x_1 &\leq 500, \quad 0.05 \leq x_2 \leq 120
 \end{aligned} \tag{13}$$

According to the obtained results in Table 12, the SNS performs better than the proposed RTH algorithm in terms of mean and StD. However, achieving the stop criteria took many iterations (5000), whereas the RTH requires only 130 iterations.*f)* **Corrugated design:** This task seeks to reduce the weight of a corrugated bulkhead in a chemical tanker⁶⁶. The optimization variables are width (x_1), depth (x_2), length (x_3), and plate thickness (x_4). Its mathematical model of this problem can be expressed as follows:

**Figure 25.** CEC 2022 unimodal function (F1) ANOVA results.**Figure 26.** CEC 2022 basic functions ANOVA results.

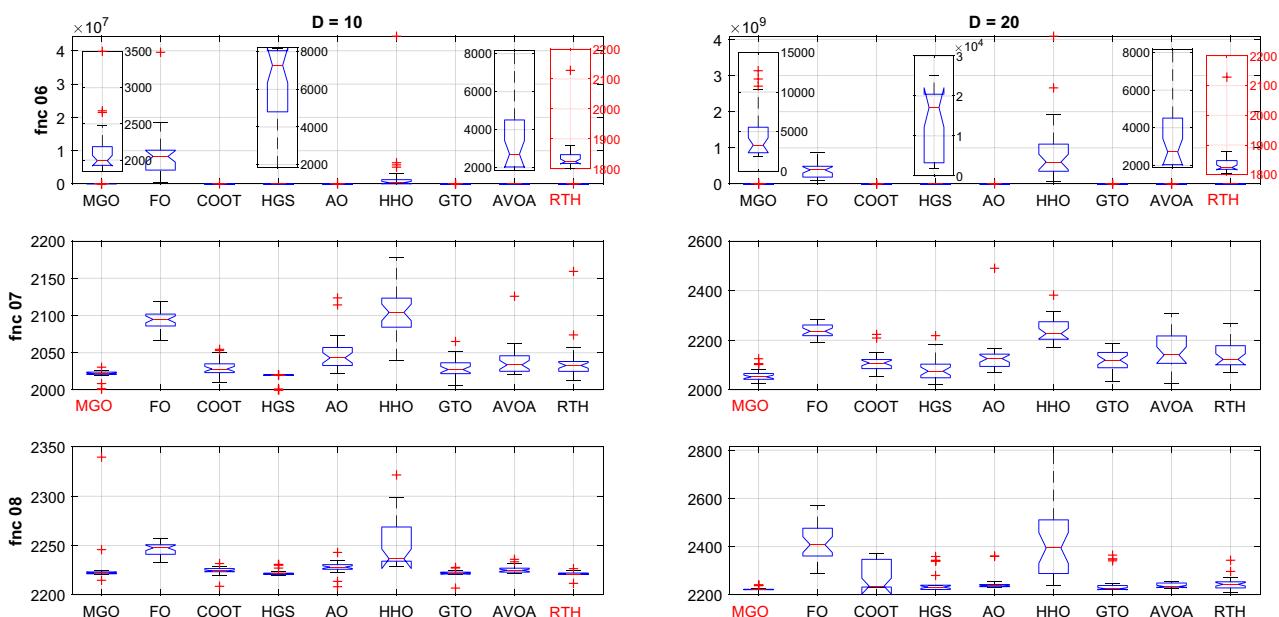


Figure 27. CEC 2022 hybrid functions ANOVA results.

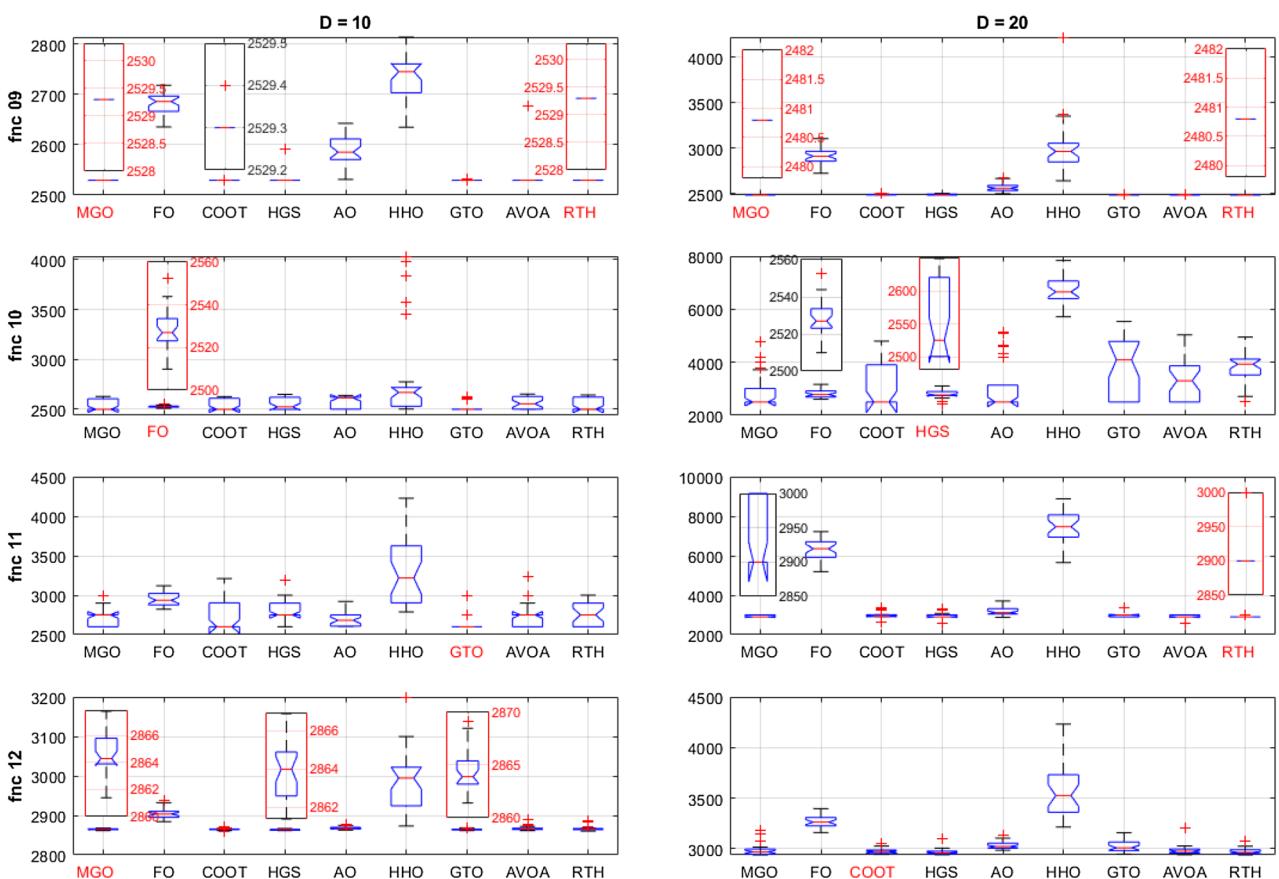
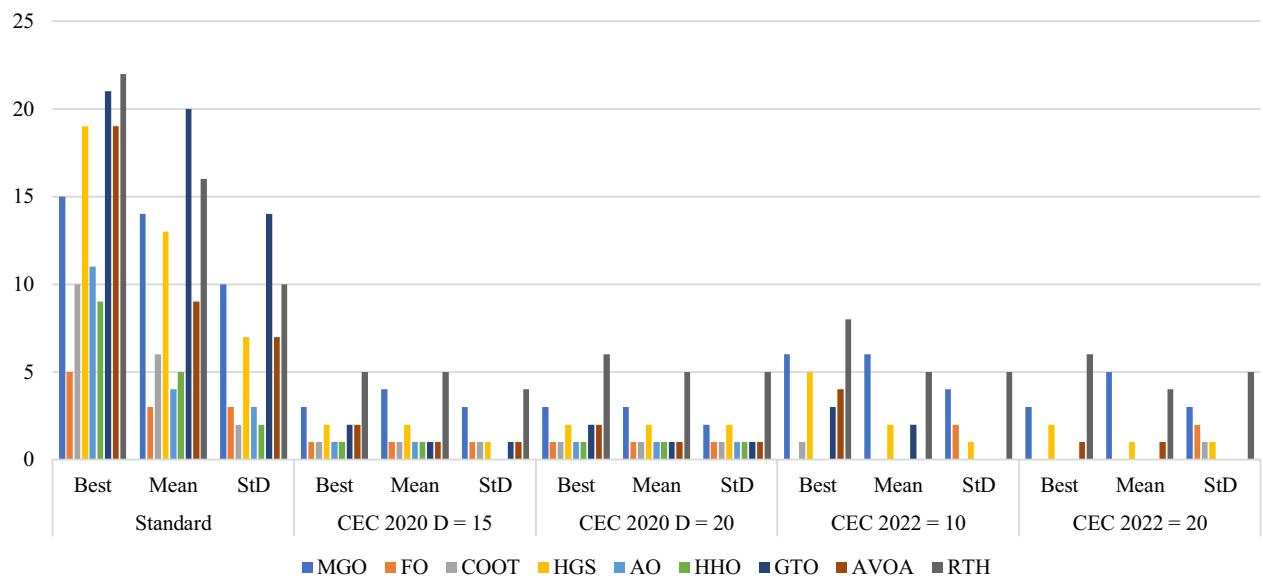
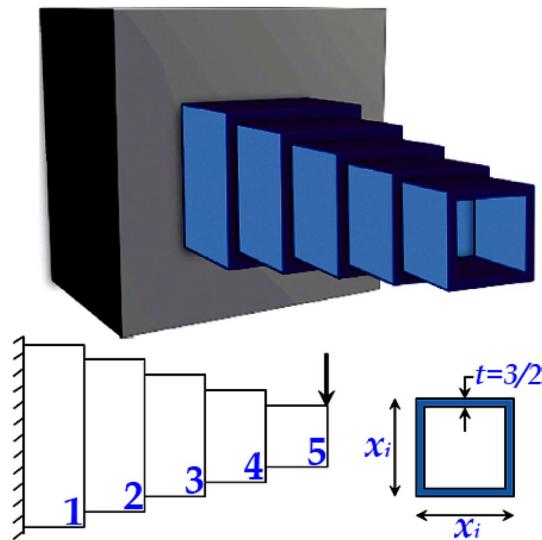


Figure 28. CEC 2022 composite functions ANOVA results.

**Figure 29.** Classes one test results.**Figure 30.** Cantilever beam schematic illustration.

MA	Worst ($\times 10^{-2}$)	Mean ($\times 10^{-2}$)	Best ($\times 10^{-2}$)	StD($\times 10^{-5}$)	Elapsed iterations
SOS ⁵⁸	NA	1.30884	1.30741	4.0	5000
CS ⁵⁹	1.35365	1.32165	1.30747	13.45	5000
AOS ⁶⁰	1.38140	1.31788	1.30741	15.55	100,000
SNS ³⁴	1.30764	1.30743	1.30741	4.31×10^{-2}	3600
PFOA ⁶¹	1.307412	1.307412	1.307412	2.05×10^{-4}	24,000
AHA ⁶²	1.343036	1.340146	1.339957	7.91×10^{-5}	15,000
FA ⁶³	1.339960	1.339957	1.339956	2.51×10^{-7}	NA
RTH	1.307410	1.307410	1.307400	1.00×10^{-8}	300

Table 8. Comparison of RTH results with the other algorithms for the I-shaped beam problem.

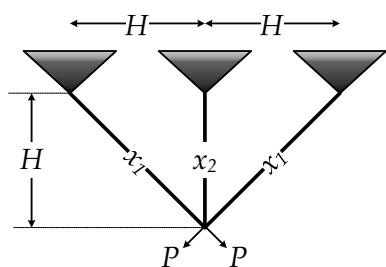


Figure 31. Three-bar truss schematic illustration.

MA	Worst	Mean	Best	StD	Elapsed iterations
CS ⁵⁹	NA	264.066900	263.9715600	9.00×10^{-5}	15,000
GWO ⁶⁴	263.904218	263.897955	263.896006	1.61×10^{-3}	50,000
WCA ⁶⁵	263.896201	263.895903	263.895843	8.71×10^{-5}	5250
PSO ⁶⁶	264.584903	263.957414	263.895843	1.37×10^{-1}	50,000
CGO ⁶⁷	263.896007	263.895851	263.895844	2.51×10^{-5}	100,000
SNS ³⁴	263.895856	263.895846	263.895843	3.31×10^{-6}	4800
PFOA ⁶¹	263.895844	263.895843	263.895842	2.01×10^{-6}	18,000
RL-BA ⁶⁸	263.924700	263.900300	263.895840	6.06×10^{-6}	NA
AHA ⁶²	263.895843	263.895843	263.895843	1.09×10^{-7}	15,000
RTH	263.895843	263.895843	263.895843	5.78×10^{-14}	670

Table 9. Comparison of RTH results with the other used algorithms for the three-bar truss design problem. The best results are marked in bold.

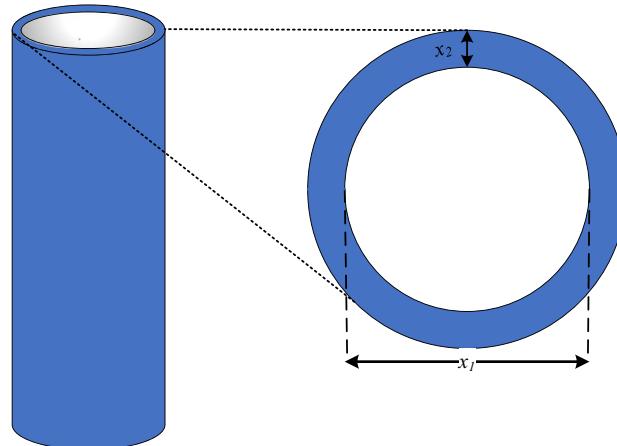


Figure 32. Tubular column schematic illustration.

MA	Worst	Mean	Best	StD	Elapsed iterations
ISA ⁶⁹	26.532	26.531	26.531	1.70×10^{-4}	3000
CS ⁵⁹	26.53972	26.53504	26.53217	1.93×10^{-3}	15,000
AOS ⁶⁰	26.60831361	26.53161399	26.53137828	1.03×10^{-3}	100,000
SNS ³⁴	26.48637095	26.48636249	26.48636147	2.22×10^{-6}	1250
PFOA ⁶¹	26.48636148	26.48636150	26.48636152	2.00×10^{-8}	24,000
AOA ⁷⁰	27.16391	26.80510	26.53730	0.02761	3000
AOA-NM ⁷⁰	26.53169	26.53133	26.53132	5.36×10^{-6}	3000
RTH	26.48636147	26.48636147	26.48636147	7.23×10^{-15}	300

Table 10. RTH results and the other used algorithms for tubular column designing problem. The best results are marked in bold.

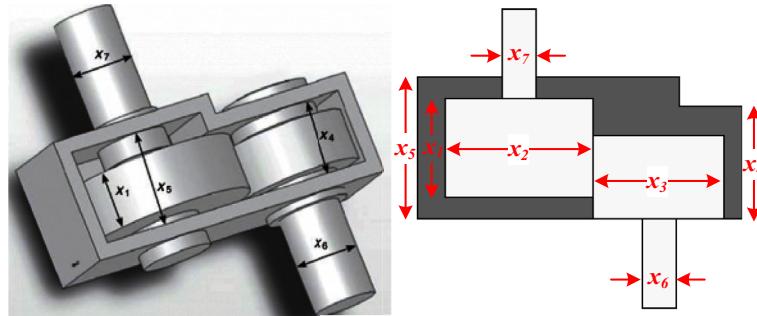


Figure 33. Speed reducer design schematic illustration.

MA	Worst	Mean	Best	StD	Elapsed iterations
CS ⁵⁹	3009.00000	3007.19970	3000.9810	4.96	250,000
ABC ⁷²	NA	2997.05841	2997.05841	0.20	30,000
WCA ⁶⁵	2994.50558	2994.47439	2994.47107	7.40×10^{-3}	15,150
APSO ⁷³	4443.01764	3822.64062	3187.63047	3.66×10^{-2}	30,000
CGO ⁶⁷	2995.50493	2994.46540	2994.44365	0.11	100,000
SNS ³⁴	2994.47110	2994.47101	2994.47107	7.00×10^{-6}	3750
AHA ⁶²	2994.47116	2994.47165	2994.47323	4.25×10^{-4}	30,000
ARSCA ⁷⁴	NA	NA	2995.821	NA	NA
RTH	2994.42400	2994.42400	2994.42400	4.63×10^{-13}	212

Table 11. Comparison of RTH results with other algorithms for speed reducer designing problems.

$$\begin{aligned}
 f(x) &= \frac{5.885x_4(x_1 + x_3)}{x_1 + \sqrt{|x_3^2 - x_2^2|}} \\
 g_1(x) &= -x_4x_2\left(0.4x_1 + \frac{x_3}{6}\right) + 8.94\left(x_1 + \sqrt{|x_3^2 - x_2^2|}\right) \leq 0 \\
 g_2(x) &= -x_4x_2^2\left(0.2x_1 + \frac{x_3}{12}\right) + 2.2\left(8.94(x_1 + \sqrt{|x_3^2 - x_2^2|})\right)^{\frac{4}{3}} \leq 0 \\
 g_3(x) &= -x_4 + 0.0156x_1 + 0.015 \leq 0 \\
 g_4(x) &= -x_4 + 0.0156x_3 + 0.015 \leq 0 \\
 g_5(x) &= -x_4 + 0.015 \leq 0 \\
 g_6(x) &= x_2 - x_3 \leq 0 \\
 0 &\leq x_1, x_2 \leq 100, x_3 \leq 100, 4 \leq x_4 \leq 5,
 \end{aligned} \tag{14}$$

The statistical results of this problem are provided in Table 13. Although AOA achieved the best results, these results were only obtained after 40,000 iterations, unlike the proposed RTH for the stop value after 129 iterations,

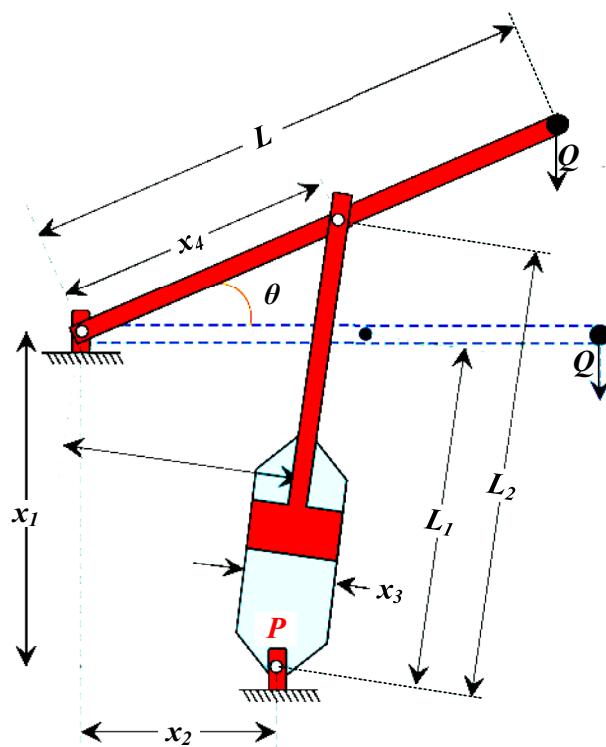


Figure 34. Piston lever design schematic illustration.

MA	Worst	Mean	Best	StD	Elapsed iterations
CS ⁵⁹	168.5920	40.2319	8.4271	59.06	50,000
PSO ⁶⁶	294	166	122	51.7	50,000
ISA ⁶⁹	610.6	226.5	8.4	111.2	12,500
AOS ⁶⁰	167.6650	33.7413	8.4191	93.47	100,000
CGO ⁶⁷	167.4728	45.0487	8.4128	67.25	100,000
SNS ³⁴	167.4728	24.3190	8.4127	47.72	5000
SSA ⁷⁵	653.4973	276.9405	8.4220	121.42	25,000
MFO ⁷⁵	167.4727	91.1239	8.4126	80.27	25,000
MVO ⁷⁵	356.2368	138.4470	8.4289	138.51	25,000
EO ⁷⁵	167.4727	100.6675	8.4127	79.30	25,000
RTH	167.4727	30.8287	8.4127	71.54	130

Table 12. Comparison of RTH results with the other used algorithms for the piston lever problem. The best results are marked in bold.

MA	Worst	Mean	Best	StD	Elapsed iterations
FA ⁷⁶	NA	10.23	7.21	1.95	12,000
LF-FA ⁷⁶	NA	8.83	6.95	1.26	12,000
LS-LF-FA ⁷⁶	NA	7.44	6.86	0.67	12,000
AD-IFA ⁷⁶	NA	7.21	6.84	0.58	12,000
AOS ⁶⁰	7.06694	7.06081	6.84296	6.49×10^{-4}	100,000
SNS ³⁴	6.84307	6.84298	6.84296	2.09×10^{-5}	3125
DMO ⁷⁷	5.9617	5.031	5.002	0.996	40,000
RTH	6.84296	6.84296	6.84295	4.51×10^{-15}	129

Table 13. Comparison of RTH results with the other used algorithms for the corrugated bulkhead design problem. The best results are marked in bold.

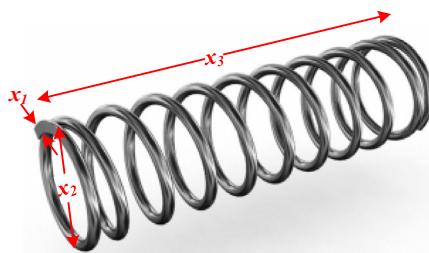


Figure 35. Speed reducer design schematic illustration.

MA	Worst	Mean	Best	StD	Elapsed iterations
PSO ⁶⁶	0.071802	0.019555	0.012857	1.17×10^{-2}	20,000
CPSO ⁷⁹	0.012924	0.012730	0.012675	5.20×10^{-5}	200,000
HPSO ⁸⁰	0.012719	0.012707	0.012665	1.58×10^{-5}	81,000
WCA ⁶⁵	0.012952	0.012746	0.012665	8.06×10^{-5}	11,750
MCEO ⁸¹	0.013509	0.012720	0.012661	3.79×10^{-5}	2000
EO ⁸²	0.013997	0.013017	0.012666	3.91×10^{-4}	15,000
SNS ³⁴	0.012766	0.012685	0.012665	2.39 × 10⁻⁵	9000
SCMWOA ⁸³	NA	0.013400	0.012670	2.40×10^{-4}	2460
RL-BA ⁶⁸	0.012928	0.012745	0.012676	7.19×10^{-4}	NA
DDAO ⁸⁴	0.017320	0.0151829	0.012907	1.26×10^{-3}	NA
RTH	0.013185	0.012776	0.012665	1.41×10^{-4}	300

Table 14. Comparison of RTH results with the other used algorithms for tension/compression spring design problem. The best results are marked in bold.

which approves its solving speed.*g) Design of tension/compression spring:* tension/compression spring design challenge aims to decrease the weight of a tension/compression spring, as detailed in⁷⁸ and Fig. 35. Minimum deflection, shear stress, surge frequency, outside diameter limitations, and design factors all play a role in this problem. The optimization variables include the mean coil diameter (x_1), the wire diameter (x_2), and the number of active coils (x_3). This problem can be expressed as follows:

$$\begin{aligned}
 f(x) &= (x_3 + 2)x_2x_1^2 \\
 g_1(x) &= 1 - \frac{x_3x_2^3}{71785x_1^4} \leq 0 \\
 g_2(x) &= \frac{x_3x_2^3}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0 \\
 g_3(x) &= 1 - \frac{140.45x_1}{x_3x_2^2} \leq 0, \quad g_4(x) = \frac{x_1 + x_2}{1.5} \leq 0 \\
 0.05 \leq x_1 &\leq 2, \quad 0.05 \leq x_1 \leq 2, \quad 0.05 \leq x_1 \leq 2,
 \end{aligned} \tag{15}$$

Table 14 compares the RTH statistical results with other MAs. The RTH algorithm solves this problem in only 130 iterations, much less than the other considered algorithms with better results.

Proton Exchange membrane fuel cell parameters' identification

Fuel cells are recent energy-generation devices that produce efficient power by directly using fuel (mainly hydrogen) and oxygen while emitting no pollutants or noise (water, heat, and electricity). Because of its benefits and wide range of applications, proton exchange membrane fuel cells (PEMFCs) that employ polymers as electrolytes are becoming more prominent^{85,86}. Fuel cell modeling is a crucial stage that enables the investigation and improvement of its performance⁸⁷. The PEMFC model is built from nonlinear differential equations that explain internal chemical processes. These equations contain various empirical parameters that must be accurately identified to improve model accuracy. As a result, identifying these parameters is essential for creating an accurate model.

PEMFC model. The PEMFC output voltage can be provided as follows⁸⁸

$$V_{fc} = E_{Nernest} - V_{act} - V_{ohm} - V_{con} \quad (16)$$

where V_{act} is the activation voltage losses, V_{ohm} is the ohmic voltage losses, V_{con} represents the concentration voltage losses, and $E_{Nernest}$ denotes the thermodynamic potential voltage (the Nernst voltage). $E_{Nernest}$ can be calculated as follows:

$$E_{Nernest} = 1.229 - 0.85 \times 10^{-3}(T - 298.15) + 4.3085 \times 10^{-5} \times T \left(\ln(P_{H_2}) + \frac{\ln(P_{O_2})}{2} \right) \quad (17)$$

where T represents the operating temperature, P_{H_2} and P_{O_2} are the partial pressures of the hydrogen and the oxygen, respectively. They can be calculated as follows:

$$P_{H_2} = 0.5 \times R_{ha} \times P_{H_2O} \left[\frac{1}{\frac{R_{ha} \times P_{H_2O}}{P_a} \times e^{\frac{1.635(i/A)}{T^{1.334}}}} - 1 \right] \quad (18)$$

$$P_{O_2} = 0.5 \times R_{hc} \times P_{H_2O} \left[\frac{1}{\frac{R_{hc} \times P_{H_2O}}{P_a} \times e^{\frac{1.635(i/A)}{T^{1.334}}}} - 1 \right] \quad (19)$$

where R_{ha} and R_{hc} represent the vapor humidities of both anode and cathode, P_a and P_c present the inlet pressures of both anode and cathode (atm), A is the electrode surface (cm^2), i is the current of the FC (A), and P_{H_2O} represents the water vapor saturation pressure(atm).

	NedStack PS6	BCS 500W	SR-12 500W
N (number of cells)	65	32	32
A (cm^2)	240	64	64
l (μm)	178	178	178
$P_{H_2}^*$ (bar)	1.0	1.0	1
$P_{O_2}^*$ (bar)	1.0	1.0	0.2095
T (K)	343	333	333
RH_a	100%		
RH_c	100%		

Table 15. The characteristics of the considered PEMFCs.

Optimization variables	ξ_1	ξ_2	ξ_3	ξ_4	λ	$R_C(\Omega)$	b (V)
Lower limit	-1.19969	0.001	3.6×10^{-5}	-2.6×10^{-4}	10	1×10^{-4}	0.0136
Upper limit	-0.8532	0.005	9.8×10^{-5}	-9.54×10^{-5}	24	8×10^{-4}	0.5

Table 16. The upper and the lower limits of the empirical parameters.

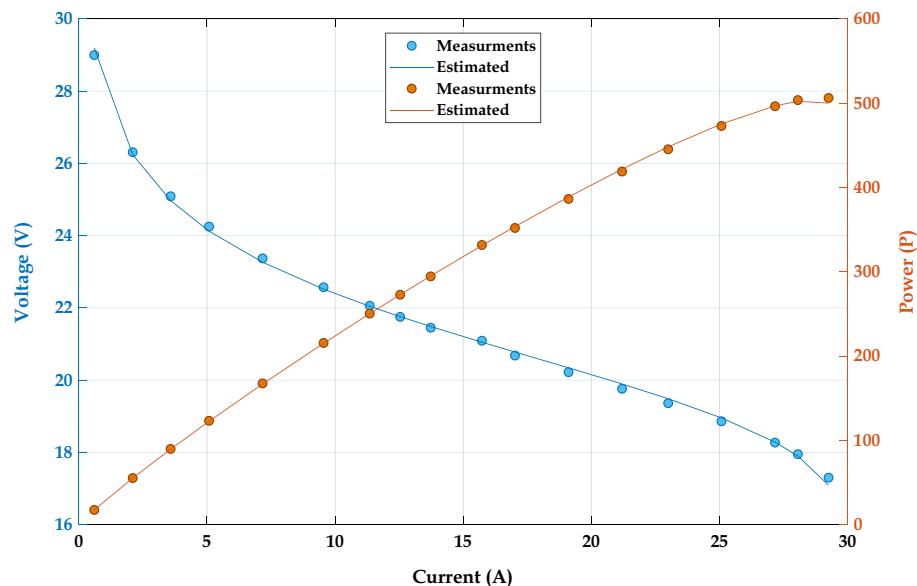
Parameters	SSA ⁸⁹	PO ⁹⁰	MPA ⁹⁰	IAEO ⁹¹	MAEO ⁹²	ISSA ⁹³	EHBO ⁹⁴	HGSA ⁹⁵	NNA ⁹⁶	RTH
ξ_1	-1.010	-1.200	-0.986	-0.810	-0.856	-1.098	-1.200	-1.11	-1.060	-1.029
$\xi_2 \times 10^{-3}$	3.220	4.042	2.609	5.17	2.73328	3.3352	3.310	3.753	3.744	3.100
$\xi_3 \times 10^{-5}$	5.450	9.800	3.600	8.790	6.634	5.903	4.200	9.710	9.690	6.487
$\xi_4 \times 10^{-5}$	-1.420	-1.929	-1.929	-1.900	-1.928	-1.928	-1.930	-1.935	-19.302	-1.936
λ	20.710	20.818	20.817	20.877	20.703	21.250	20.877	21.970	20.877	22.02
$R_C(\Omega) \times 10^{-4}$	0.075	0.016	0.016	0.161	0.100	0.161	0.100	0.100	0.100	0.100
b (V) $\times 10^{-2}$	1.00	1	1	1.00	1.60	148.2	1.60	1.60	1.60	1.740
Best	1.219	1.156	1.156	1.16	1.157	1.16	1.170	1.169	1.1698	1.140
Worst	1.520	NA	NA	1.17	NA	179.26	1.185	1.34	1.3670	1.140
Mean	NA	1.88	1.16	1.16	2.9326	14.57	1.174	NA	NA	1.140
StD	871×10^{-2}	NA	NA	144×10^{-3}	NA	NA	0.6×10^{-4}	3.4×10^{-4}	5.64×10^{-4}	9.37×10^{-8}

Table 17. The extracted BSC 500W FC parameters. The best results are marked in bold.

	SSA ⁸⁹	EO ⁹⁷	STSA ⁹⁸	IAEO ⁹¹	EBHO ⁹⁴	mAEFA ⁹⁹	NNA ⁹⁶	RTH
ξ_1	1.130	-1.12171	-0.853	-1.1997	-0.85396	-1.149	-0.8535	-0.90568
$\xi_2 \times 10^{-5}$	3.460	3.77	2.840	3.4103	2.40	3.349	2.4316	3.44
$\xi_3 \times 10^{-5}$	4.590	7.81	6.790	3.60	3.60	3.60	3.7545	8.76
$\xi_4 \times 10^{-5}$	-9.620	-9.54	-9.540	-9.54	-9.54	-9.5	-9.5400	-9
λ	12.910	16.60171	13.463	19.7903	13.465	13.097512	13.0802	17.80574
$R_c \times 10^{-3}$	0.100	0.205	0.100	0.362	0.1	0.1	0.1000	8
$b \times 10^{-2}$	6.000	0.0285	1.360	1.360	1.360	1.360	1.360	8.185
Best	2.181	2.40931	2.146	2.1459	2.14570	2.07974	2.14487	2.1058
Worst	2.251	3.02680	3.183	2.1459	2.14570	2.08019	2.1645	2.1058
Mean	NA	2.61680	0.280	2.1459	2.14570	2.07987	NA	2.1058
StD	0.020	0.148	0.177	NA	5.69×10^3	1.6×10^{-4}	5.848×10^3	1.79×10^{-6}

Table 18. The extracted NedStack PS6 FC parameters. The best results are marked in bold.

Parameters	PO ⁹⁰	LSHADE ¹⁰⁰	MPA ⁹⁰	VSDE ¹⁰¹	MAEO ⁹²	ISSA ⁹³	TGA ¹⁰²	RTH
ξ_1	-0.860	1.216	-1.028	-0.952	-0.860	-159	-1.112	-0.906
$\xi_2 \times 10^{-3}$	3.376	-0.960	3.898	3.000	2.771	4.146	3.855	3.440
$\xi_3 \times 10^{-5}$	9.794	2.621	9.800	7.783	6.170	5.6443	4.370	8.760
$\xi_4 \times 10^{-5}$	-0.954	3.60	-0.954	-2.00	-0.954	-2.2908	-0.964	-9.00
λ	23.00	-9.54	23.00	20.29	22.99	13.78	23.00	17.806
$R_C(\Omega) \times 10^{-4}$	6.723	0.154	6.723	1.00	6.707	1.00	2.19	8.00
$b(V) \times 10^{-2}$	17.5	23.999	17.5	2.79	17.5	7.4	18.3	8.185
Best	1.057	1.216	1.057	1.0526	1.057	0.792	1.104	0.5607
Worst	NA	3.508	NA	1.1875	NA	1.793	5.504	0.5607
Mean	1.058	3.001	1.057	1.0834	6.43	1.46	2.064	0.5607
StD	NA	0.12402	NA	0.1768	NA	NA	NA	1.79×10^{-6}

Table 19. The extracted SR-12 500W FC parameters. The best results are marked in bold.**Figure 36.** Voltage and power curves of BCS 500W using the proposed RTH algorithm.

The activation voltage losses (V_{act}) can be obtained as

$$V_{act} = -(\xi_1 + \xi_2 T + \xi_3 T \ln(C_{O_2}) + \xi_4 T \ln(i)) \quad (20)$$

where $\xi_1, \xi_2, \xi_3, \xi_4$ denote semi-empirical parameters; C_{O_2} is the concentration of oxygen at the cathode's surface (mol.cm⁻³). It can be calculated as follows:

$$C_{O_2} = \frac{P_{O_2}}{5.08 \times 10^6} e^{\left(\frac{498}{T} \right)} \quad (21)$$

The ohmic losses (V_{ohm}) can be obtained as

$$V_{ohm} = i(R_m + R_c) \quad (22)$$

where R_c is the resistance of the connectors, and R_m is the resistance of the membrane. R_m can be calculated as

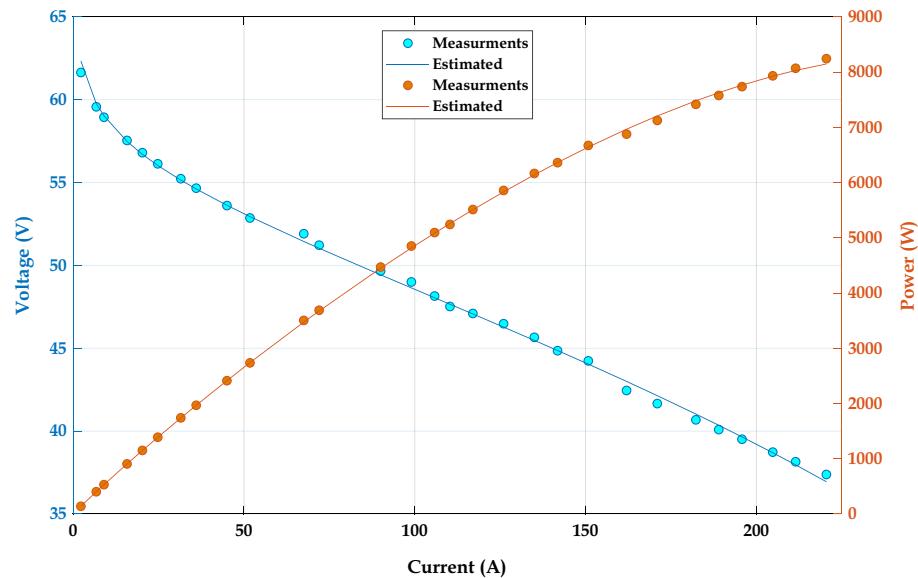


Figure 37. Voltage and power curves of BCS 500W using the proposed RTH algorithm.

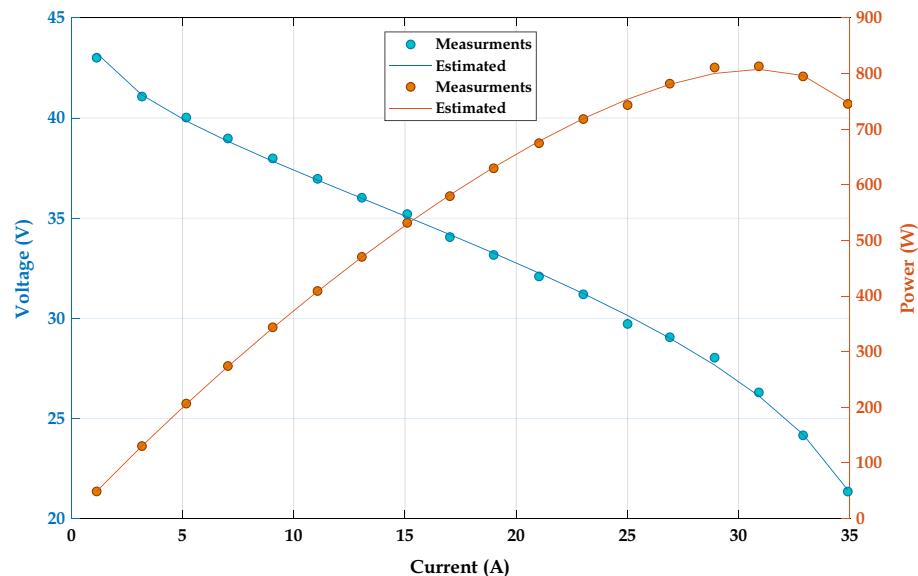


Figure 38. Voltage and power curves of SR-12 500W using the proposed RTH algorithm.

$$R_m = \rho_m \left(\frac{l}{A_m} \right) \quad (23)$$

where l and A_m represent the membrane thickness (cm) and surface (cm^2), respectively, and ρ_m represents the membrane-specific resistivity ($\text{ohm} \times \text{cm}$). ρ_m can be obtained as

$$\rho_m = \frac{181.6 \left[1 + 0.03i/A_m + 0.062T/303 \left(i/A_m \right)^{2.5} \right]}{\left[\lambda - 0.634 - 3i/A_m \right] e^{4.18 \frac{T-303}{T}}} \quad (24)$$

where λ is the membrane material's water content.

Objective function. The sum square error (SSE) between the measured (V_{data}) and model output data (V_{model}) will be used as an objective function. The objective function can be constructed as follows:

$$f(x) = \sum_{k=1}^N (V_{\text{data}}(k) - V_{\text{model}}(k, x))^2 \quad (25)$$

where N represents the data size, and x represents a vector containing seven unknown parameters.

$$x = [\zeta_1 \ \zeta_2 \ \zeta_3 \ \zeta_4 \ R_c \ \lambda \ b] \quad (26)$$

The FC data are compared with those generated by the model, and the fitness value is calculated based on the error between them. The model is developed in MATLAB script, and the data are loaded from the Excel sheet file. The identification is an iterative process that updates the candidate solutions at each iteration by sending them to the MATLAB script that includes the FC model and simulates it after that, and then generates the fitness value. This process repeated until the last iteration.

Results. The suggested RTH algorithm will be used to extract the seven unknown characteristics of three PEM fuel cells: the NedStack PS6, the BCS 500W, and the SR-12 500W. Table 15³⁷ provided the accurate values of the parameters, testing operating conditions, and measurement data for the tested PEMFC types. The upper and lower limits of the empirical parameters are presented in Table 16. The results are compared to those published for the same FC types.

Table 17 shows the comparative findings for the BCS 500 W type, Table 18 for the NedStack PS6, and Table 19 for the SR-12 PEM 500 W. The comparison findings are primarily based on sum square error (SSE).

Based on the findings reported in Table 17, the suggested method-based RTH performed the best of all the reported techniques in this comparison to extract the parameters of the BCS 500W. The proposed RTH algorithms obtained the minimal fitness function (SSE) by 1.14×10^{-2} . On the other hand, the mean fitness values were similar to the best value. The RTH provides the lowest STD values compared to the different algorithms (9.37×10^{-8}), indicating its robustness. Figure 36 depicts the experimental and estimated voltage and power curves of BCS 500W using the suggested RTH algorithm. The estimated voltage and power curves match the experimental curves. These curves demonstrate the proposed algorithm's accuracy in deriving the best BCS 500W parameters.

Similar to the BDC 500W case, the suggested method-based RTH performed the best of all the cited methods in this comparison to extract the parameters of the NedStack PS6. The proposed RTH best result is 2.058, similar to the mean values. The RTH provides the lowest StD values compared to the different algorithms (1.79×10^{-6}), approving its robustness. Figure 37 depicts the experimental and estimated voltage and power curves of NedStack PS6 using the suggested RTH algorithm. The estimated voltage and power curves match the experimental curves.

Similar to the previous cases, the suggested method-based RTH performed the best of all the cited methods in this comparison to extract the parameters of the SR 500W. The proposed RTH best result is 0.5607, similar to the mean values. The RTH also provides the lowest StD value by 1.79×10^{-6} , approving its robustness. Figure 38 shwos the experimental and estimated voltage and power curves of SR 500W using the suggested RTH algorithm.

The obtained results for the standard test functions evaluated the performance of the proposed RTH algorithm for both exploitation and exploration phases compared to other MAs. From these results, the proposed RTH has excellent exploitation and exploration abilities. In addition, its convergence speed has been approved. Then, the proposed RTH efficiently has been evaluated for several real-world applications. Its performance has been compared to several published algorithms for each problem. The achieved results confirm its ability to handle these problems. Finally, a PEMFC parameters extraction has been performed using the proposed RTH algorithm. The results have been compared to other well-known algorithms to approve its performance. The results achieved were excellent. Hence, its performance has been elaborated and approved for various types of optimization problems.

Conclusion and future works

This paper has proposed a novel metaheuristic optimization algorithm named the red-tailed hawk (RTH) algorithm to solve various optimization tasks and problems. The proposed RTH is inspired by the red-tailed hawk's hunting behaviors of a predatory bird. A mathematical model has been developed to replicate the behavior of red-tailed hawks. The main contribution of this paper is to propose a new optimizer that has high robustness and fast convergence speed when solving various optimization problems. RTH's performance was firstly evaluated

using three types of mathematical functions that express the nature of different optimization problems: twenty-three standard benchmark test functions, IEEE Congress on Evolutionary Computation 2020 (CEC2020) with 15 and 20 search space dimensions, and CEC2022 with 10 and 20 search space dimensions. These functions enable evaluating the exploitative ability, exploratory ability, and local optima avoidance of RTH. The results are compared to other recent and robust optimizers, including Farmland Fertility Optimizer (FO), African Vultures Optimization Algorithm (AVOA), Mountain Gazelle Optimizer (MGO), Gorilla Troops Optimizer (GTO), COOT algorithm, Hunger Games Search (HGS), Aquila Optimizer (AO), and Harris Hawks optimization (HHO). The results show that the proposed algorithm can provide the optimal solution for most of the considered functions with fast convergence speed and good robustness. Then, the findings of the seven constrained engineering design problems demonstrated that the RTH could show superior results to other published algorithms in terms of precision, robustness, and convergence rate. To deeply investigate the performance of the proposed RTH, the results of the proton exchange membrane fuel cell parameters extractions (PEMFC) have used the proposed RTH algorithm compared to published ones. The ultimate results show the RTH's ability to find better parameters for the dynamic model of the PEMFC.

RTH's performance is anticipated to be considerably improved by integrating more complicated processes and combining effective operators and technics of other heuristics. Enhancing the proposed algorithm by including other factors in the model, such as the wind effect and prey escaping, is possible. However, the bigger code size of the RTH compared to other algorism like the PSO and the SSA can be a problem for its implementation. Bus, this problem can be bypassed with the utilization of fast calculators.

Data availability

All data generated or analyzed during this study are included in this published article and its supplementary information files.

Received: 13 March 2023; Accepted: 14 July 2023

Published online: 09 August 2023

References

- Mahadeva, R., Kumar, M., Gupta, V., Manik, G. & Patole, S. P. Modified Whale Optimization Algorithm based ANN: a novel predictive model for RO desalination plant. *Sci. Rep.* **13**, 2901 (2023).
- Zhang, C. & Ordóñez, R. Numerical optimization. *Adv. Ind. Control* https://doi.org/10.1007/978-1-4471-2224-1_2 (2012).
- Wu, G. Across neighborhood search for numerical optimization. *Inf. Sci.* **329**, 597–618 (2016).
- Zhao, W., Wang, L. & Zhang, Z. Artificial ecosystem-based optimization: A novel nature-inspired meta-heuristic algorithm. *Neural Comput. Appl.* **32**, 9383–9425 (2020).
- Wang, L., Zhao, W., Tian, Y. & Pan, G. A bare bones bacterial foraging optimization algorithm. *Cogn. Syst. Res.* **52**, 301–311 (2018).
- Sharma, S., Saha, A. K. & Lohar, G. Optimization of weight and cost of cantilever retaining wall by a hybrid metaheuristic algorithm. *Eng. Comput.* **38**, 2897–2923 (2022).
- Sharma, S., Saha, A. K., Majumder, A. & Nama, S. MPBOA: A novel hybrid butterfly optimization algorithm with symbiosis organisms search for global optimization and image segmentation. *Multimed. Tools Appl.* **80**, 12035–12076 (2021).
- Talbi, E.-G. *Metaheuristics From Design to Implementation* (Wiley, 2009).
- Drexl, J., Petrowski, A., Siarry, P., Taillard, E. & Chatterjee, A. *Metaheuristics for Hard Optimization: Methods and Case Studies*. (2006).
- Heidari, A. A., Abbaspour, R. A. & Jordehi, A. R. An efficient chaotic water cycle algorithm for optimization tasks. *Neural Comput. Appl.* **28**, 1–29 (2015).
- Mafarja, M. *et al.* Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems. *Knowledge-Based Syst.* **145**, 25–45 (2018).
- Salcedo-Sanz, S. Modern meta-heuristics based on nonlinear physics processes: A review of models and design procedures. *Phys. Rep.* **655**, 1–70 (2016).
- Hosseini-Gandomi, A., Xin-She, Y. & Hosseini-Alavi, A. Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems. *Eng. Comput.* **29**, 17–35 (2013).
- Batinu, F., Carpaneto, E. & Chicco, G. A unified scheme for testing alternative techniques for distribution system minimum loss reconfiguration. in *2005 International Conference on Future Power Systems* vol. 2005 6 pp. (IEEE, 2005).
- Shayanfar, H. & Gharehchopogh, F. S. Farmland fertility: A new metaheuristic algorithm for solving continuous optimization problems. *Appl. Soft Comput.* **71**, 728–746 (2018).
- Abdollahzadeh, B., Gharehchopogh, F. S. & Mirjalili, S. African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems. *Comput. Ind. Eng.* **158**, 107408 (2021).
- Abdollahzadeh, B., Gharehchopogh, F. S., Khodadadi, N. & Mirjalili, S. Mountain Gazelle Optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems. *Adv. Eng. Softw.* **174**, 103282 (2022).
- Abdollahzadeh, B., Soleimanian Gharehchopogh, F. & Mirjalili, S. Artificial gorilla troops optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems. *Int. J. Intell. Syst.* **36**, 5887–5958 (2021).
- Naruei, I. & Keynia, F. A new optimization method based on COOT bird natural life model. *Expert Syst. Appl.* **183**, 115352 (2021).
- Yang, Y., Chen, H., Heidari, A. A. & Gandomi, A. H. Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts. *Expert Syst. Appl.* **177**, 114864 (2021).
- Abualigah, L. *et al.* Aquila Optimizer: A novel meta-heuristic optimization algorithm. *Comput. Ind. Eng.* **157**, 107250 (2021).
- Heidari, A. A. *et al.* Harris hawks optimization: Algorithm and applications. *Futur. Gener. Comput. Syst.* **97**, 849–872 (2019).
- Mirjalili, S. & Lewis, A. The Whale optimization algorithm. *Adv. Eng. Softw.* **95**, 51–67 (2016).
- Faris, H. *et al.* An efficient binary Salp Swarm algorithm with crossover scheme for feature selection problems. *Knowledge-Based Syst.* **154**, 43–67 (2018).
- Bian, F., Li, T. & Cong, P. Genetic programming. *Fenxi Huaxue* **26**, 783–785 (1998).
- Lapre, C. *et al.* Genetic algorithm optimization of broadband operation in a noise-like pulse fiber laser. *Sci. Rep.* **13**, 1865 (2023).
- Storn, R. & Price, K. Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **11**, 341–359 (1997).

28. Zheng, Y., Lu, X., Zhang, M. & Chen, S. Biogeography-based optimization. In *Biogeography-Based Optimization: Algorithms and Applications* 27–49 (Springer, 2019). https://doi.org/10.1007/978-981-13-2586-1_2.
29. Bai, H., Cao, Q. & An, S. Mind evolutionary algorithm optimization in the prediction of satellite clock bias using the back propagation neural network. *Sci. Rep.* **13**, 2095 (2023).
30. Rashedi, E., Nezamabadi-pour, H. & Saryazdi, S. GSA: A gravitational search algorithm. *Inf. Sci.* **179**, 2232–2248 (2009).
31. Ismaael, A. A. K., Houssein, E. H., Oliva, D. & Said, M. Gradient-based optimizer for parameter extraction in photovoltaic models. *IEEE Access* **9**, 13403–13416 (2021).
32. Azizi, M., Aickelin, U., Khorshidi, A. & H. & Baghalzadeh Shishehgarkhaneh, M., Energy valley optimizer: a novel metaheuristic algorithm for global and engineering optimization. *Sci. Rep.* **13**, 226 (2023).
33. Kumar, M., Kulkarni, A. J. & Satapathy, S. C. Socio evolution & learning optimization algorithm: A socio-inspired optimization methodology. *Futur. Gener. Comput. Syst.* **81**, 252–272 (2018).
34. Bayzidi, H., Talatahari, S., Saraei, M. & Lamarche, C.-P. Social network search for solving engineering optimization problems. *Comput. Intell. Neurosci.* **2021**, 1–32 (2021).
35. Verirkazemi, M. & Fazeli Veysari, E. A new optimization algorithm inspired by the quest for the evolution of human society: Human felicity algorithm. *Expert Syst. Appl.* **193**, 116468 (2022).
36. Baykasoglu, A. & Ozsoydan, F. B. Evolutionary and population-based methods versus constructive search strategies in dynamic combinatorial optimization. *Inf. Sci. (Ny)* **420**, 159–183 (2017).
37. Kennedy, J. & Eberhart, R. Particle swarm optimization. in *Proceedings of ICNN'95—International Conference on Neural Networks* vol. 4 1942–1948 (IEEE, 1995).
38. Mirjalili, S. et al. Salp Swarm algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **114**, 163–191 (2017).
39. Chou, J.-S. & Molla, A. Recent advances in use of bio-inspired jellyfish search algorithm for solving optimization problems. *Sci. Rep.* **12**, 19157 (2022).
40. Ghafori, S. & Gharehchopogh, F. S. Advances in spotted hyena optimizer: A comprehensive survey. *Arch. Comput. Methods Eng.* **29**, 1569–1590 (2022).
41. Gharehchopogh, F. S. Advances in tree seed algorithm: A comprehensive survey. *Arch. Comput. Methods Eng.* **29**, 3281–3304 (2022).
42. Gharehchopogh, F. S., Namazi, M., Ebrahimi, L. & Abdollahzadeh, B. Advances in sparrow search algorithm: A Comprehensive Survey. *Arch. Comput. Methods Eng.* **30**, 427–455 (2023).
43. Chakraborty, P., Sharma, S. & Saha, A. K. Convergence analysis of butterfly optimization algorithm. *Soft Comput.* **27**, 7245–7257 (2023).
44. Sharma, S., Khodadadi, N., Saha, A. K., Gharehchopogh, F. S. & Mirjalili, S. Non-dominated sorting advanced butterfly optimization algorithm for multi-objective problems. *J. Bionic Eng.* **20**, 819–843 (2023).
45. Sharma, S., Chakraborty, S., Saha, A. K., Nama, S. & Sahoo, S. K. mLBOA: A Modified Butterfly optimization algorithm with lagrange interpolation for global optimization. *J. Bionic Eng.* **19**, 1161–1176 (2022).
46. Chakraborty, S., Sharma, S., Saha, A. K. & Saha, A. A novel improved whale optimization algorithm to solve numerical optimization and real-world applications. *Artif. Intell. Rev.* **55**, 4605–4716 (2022).
47. Sahoo, S. K., Sharma, S. & Saha, A. K. A novel variant of moth flame optimizer for higher dimensional optimization problems. *J. Bionic Eng.* <https://doi.org/10.1007/s42235-023-00357-7> (2023).
48. Gharehchopogh, F. S. Quantum-inspired metaheuristic algorithms: comprehensive survey and classification. *Artif. Intell. Rev.* **56**, 5479–5543 (2023).
49. Preston, C. R. *Wild Bird Guide: Red-Tailed Hawk (Wild Bird Guides)* (Stackpole Books, 2000).
50. Pearlstone, E. V. & Thompson, D. B. Geographic variation in morphology of four species of migratory raptors. *J. Raptor Res.* **38**, 334–342 (2004).
51. Ballam, J. M. The use of soaring by the red-tailed Hawk (*Buteo jamaicensis*). *Auk* **101**, 519–524 (1984).
52. Day, L. The City Naturalist: Red Tailed Hawk. in *The 79th Street Boat Basin Flora and Fauna Society* (2007).
53. Knight, R. L., Andersen, D. E., Bechard, M. J. & Marr, N. V. Geographic variation in nest-defence behaviour of the Red-tailed Hawk *Buteo jamaicensis*. *Ibis* **131**, 22–26 (2008).
54. Yang, X.-S. Appendix A: Test Problems in Optimization. In *Engineering Optimization* 261–266 (Wiley, 2010). <https://doi.org/10.1002/9780470640425.app1>.
55. Yue, CT and Price, KV and Suganthan, Ponnuthurai N and Liang, JJ and Ali, Mostafa Z and Qu, BY and Awad, Noor H and Biswas, P. P. Problem definitions and evaluation criteria for the CEC 2020 special session and competition on single objective bound constrained numerical optimization. *Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China, Tech. Rep* **201911**, (2019).
56. Suganthan, P. N. et al. *Problem definitions and evaluation criteria for the CEC 2021 special session and competition on single objective bound constrained numerical optimization*. Technical Report, Nanyang Technological University, Singapore, May 2005 AND KanGAL Report 2005005, IIT Kanpur, India (2020).
57. Faramarzi, A., Heidarinejad, M., Mirjalili, S. & Gandomi, A. H. Marine Predators Algorithm: A nature-inspired metaheuristic. *Expert Syst. Appl.* **152**, 113377 (2020).
58. Cheng, M.-Y. & Prayogo, D. Symbiotic organisms search: A new metaheuristic optimization algorithm. *Comput. Struct.* **139**, 98–112 (2014).
59. Kaveh, A. *Advances in Metaheuristic Algorithms for Optimal Design of Structures* (Springer, 2021). <https://doi.org/10.1007/978-3-030-59392-6>.
60. Azizi, M. Atomic orbital search: A novel metaheuristic algorithm. *Appl. Math. Model.* **93**, 657–683 (2021).
61. Cao, S. et al. A novel meta-heuristic algorithm for numerical and engineering optimization problems: Piranha foraging optimization algorithm (PFOA). *IEEE Access* <https://doi.org/10.1109/ACCESS.2023.3267110> (2023).
62. Zhao, W., Wang, L. & Mirjalili, S. Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications. *Comput. Methods Appl. Mech. Eng.* **388**, 114194 (2022).
63. Ghasemi, M. et al. A new firefly algorithm with improved global exploration and convergence with application to engineering optimization. *Decis. Anal. J.* **5**, 100125 (2022).
64. Mirjalili, S., Mirjalili, S. M. & Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **69**, 46–61 (2014).
65. Eskandar, H., Sadollah, A., Bahreininejad, A. & Hamdi, M. Water cycle algorithm: A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Comput. Struct.* **110–111**, 151–166 (2012).
66. Ravindran, A., Ragsdell, K. M. & Reklaitis, G. V. *Engineering Optimization* (Wiley, 2006). <https://doi.org/10.1002/9780470117811>.
67. Talatahari, S. & Azizi, M. Optimization of constrained mathematical and engineering design problems using chaos game optimization. *Comput. Ind. Eng.* **145**, 106560 (2020).
68. Meng, X. B., Li, H. X. & Gao, X. Z. An adaptive reinforcement learning-based bat algorithm for structural design problems. *Int. J. Bio-Inspired Comput.* **14**, 114–124 (2019).
69. Gandomi, A. H. & Roke, D. A. Engineering optimization using interior search algorithm. In *2014 IEEE Symposium on Swarm Intelligence* 1–7 (IEEE, 2014). <https://doi.org/10.1109/SIS.2014.7011771>.

70. Yildiz, B. S. *et al.* A novel hybrid arithmetic optimization algorithm for solving constrained optimization problems. *Knowledge-Based Syst.* **271**, 110554 (2023).
71. Sattar, D. & Salim, R. A smart metaheuristic algorithm for solving engineering problems. *Eng. Comput.* **37**, 2389–2417 (2021).
72. Akay, B. & Karaboga, D. Artificial bee colony algorithm for large-scale problems and engineering design optimization. *J. Intell. Manuf.* **23**, 1001–1014 (2012).
73. Ben Guedria, N. Improved accelerated PSO algorithm for mechanical engineering optimization problems. *Appl. Soft Comput.* **40**, 455–467 (2016).
74. Yang, X. *et al.* An adaptive quadratic interpolation and rounding mechanism sine cosine algorithm with application to constrained engineering optimization problems. *Expert Syst. Appl.* **213**, 119041 (2023).
75. Gupta, S. *et al.* Comparison of metaheuristic optimization algorithms for solving constrained mechanical design optimization problems. *Expert Syst. Appl.* **183**, 115351 (2021).
76. Wu, J. *et al.* An improved firefly algorithm for global continuous optimization problems. *Expert Syst. Appl.* **149**, 113340 (2020).
77. Agushaka, J. O. *et al.* Improved Dwarf Mongoose optimization for constrained engineering design problems. *J. Bionic Eng.* **20**, 1263–1295 (2023).
78. Arora, J. S. *Introduction to Optimum Design* (Elsevier, 2012). <https://doi.org/10.1016/C2009-0-61700-1>.
79. He, Q. & Wang, L. An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Eng. Appl. Artif. Intell.* **20**, 89–99 (2007).
80. He, Q. & Wang, L. A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. *Appl. Math. Comput.* **186**, 1407–1422 (2007).
81. MiarNaeimi, F., Azizyan, G. & Rashki, M. Multi-level cross entropy optimizer (MCEO): An evolutionary optimization algorithm for engineering problems. *Eng. Comput.* **34**, 719–739 (2018).
82. Faramarzi, A., Heidarnejad, M., Stephens, B. & Mirjalili, S. Equilibrium optimizer: A novel optimization algorithm. *Knowledge-Based Syst.* **191**, 105190 (2020).
83. El-Kenawy, E.-S.M. *et al.* Novel meta-heuristic algorithm for feature selection, unconstrained functions and engineering problems. *IEEE Access* **10**, 40536–40555 (2022).
84. Ghafil, H. N. & Jármai, K. Dynamic differential annealed optimization: New metaheuristic optimization algorithm for engineering applications. *Appl. Soft Comput.* **93**, 106392 (2020).
85. Olabi, A. G., Wilberforce, T. & Abdelkareem, M. A. Fuel cell application in the automotive industry and future perspective. *Energy* **214**, 118955 (2021).
86. Rezk, H. *et al.* Fuel cell as an effective energy storage in reverse osmosis desalination plant powered by photovoltaic system. *Energy* **175**, 423–433 (2019).
87. Rezk, H. *et al.* Optimal parameter estimation strategy of PEM fuel cell using gradient-based optimizer. *Energy* **239**, 122096 (2022).
88. Cao, Y., Li, Y., Zhang, G., Jermsittiparsert, K. & Nasseri, M. An efficient terminal voltage control for PEMFC based on an improved version of whale optimization algorithm. *Energy Rep.* **6**, 530–542 (2020).
89. El-Fergany, A. A. Extracting optimal parameters of PEM fuel cells using Sarp Swarm optimizer. *Renew. Energy* **119**, 641–648 (2018).
90. Zaki Diab, A. A. *et al.* Fuel cell parameters estimation via marine predators and political optimizers. *IEEE Access* **8**, 166998–167018 (2020).
91. Rizk-Allah, R. M. & El-Fergany, A. A. Artificial ecosystem optimizer for parameters identification of proton exchange membrane fuel cells model. *Int. J. Hydrogen Energy* **46**, 37612–37627 (2021).
92. Menesy, S. *et al.* Effective parameter extraction of different polymer electrolyte membrane fuel cell stack models using a modified artificial ecosystem optimization algorithm. *IEEE Access* **8**, 31892–31909 (2020).
93. Sultan, H. M., Menesy, A. S., Kamel, S., Selim, A. & Jurado, F. Parameter identification of proton exchange membrane fuel cells using an improved salp swarm algorithm. *Energy Convers. Manag.* **224**, 113341 (2020).
94. Abdel-Basset, M., Mohamed, R., Elhoseny, M., Chakrabortty, R. K. & Ryan, M. J. An efficient heap-based optimization algorithm for parameters identification of proton exchange membrane fuel cells model: Analysis and case studies. *Int. J. Hydrogen Energy* **46**, 11908–11925 (2021).
95. Fahim, S. R. *et al.* Parameter identification of proton exchange membrane fuel cell based on hunger games search algorithm. *Energies* **14**, 5022 (2021).
96. Fawzi, M., El-Fergany, A. A. & Hasanien, H. M. Effective methodology based on neural network optimizer for extracting model parameters of PEM fuel cells. *Int. J. Energy Res.* **43**, 8136–8147 (2019).
97. Menesy, A. S., Sultan, H. M. & Kamel, S. Extracting Model Parameters of Proton Exchange Membrane Fuel Cell Using Equilibrium Optimized Algorithm. In *2020 International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE)* 1–7 (IEEE, 2020). doi:<https://doi.org/10.1109/REEPE49198.2020.9059219>.
98. Jiang, J., Xu, M., Meng, X. & Li, K. STSA: A sine Tree-Seed Algorithm for complex continuous optimization problems. *Phys. A Stat. Mech. its Appl.* **537**, 122802 (2020).
99. Houssein, E. H., Hashim, F. A., Ferahtia, S. & Rezk, H. An efficient modified artificial electric field algorithm for solving optimization problems and parameter estimation of fuel cell. *Int. J. Energy Res.* **45**, 20199–20218 (2021).
100. Fathy, A., Abdel Aleem, S. H. E. & Rezk, H. A novel approach for PEM fuel cell parameter estimation using LSHADE-EpSin optimization algorithm. *Int. J. Energy Res.* **45**, 6922–6942 (2021).
101. Fathy, A., Elaziz, M. A. & Alharbi, A. G. A novel approach based on hybrid vortex search algorithm and differential evolution for identifying the optimal parameters of PEM fuel cell. *Renew. Energy* **146**, 1833–1845 (2020).
102. Kamel, S., Jurado, F., Sultan, H. & Menesy, A. Tree growth algorithm for parameter identification of proton exchange membrane fuel cell models. *Int. J. Interact. Multimed. Artif. Intell.* **6**, 11 (2020).

Author contributions

The authors confirm their contribution to the paper as follows: study conception and design: S.F., A.D.; data collection: R.H., S.M.; analysis and interpretation of results: A.H., M.M. M.A.; draft manuscript preparation: S.F., R.H. All authors reviewed the results and approved the final version of the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1038/s41598-023-38778-3>.

Correspondence and requests for materials should be addressed to S.M.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2023