

## Nonlinear-based chaotic Harris Hawks Optimizer: algorithm and internet of vehicles application

Item Type	Journal article
Authors	Abdollahi Dehkordi, Amin;Sadiq, Ali Safaa;Mirjalili, Seyedali;Ghafoor, Kayhan Zrar
Citation	Dekhordi, A.A., Sadiq, A.S., Mirjalili, S. and Ghafoor, K.Z. (2021) Nonlinear-based chaotic Harris Hawks Optimizer: algorithm and internet of vehicles application. Applied Soft Computing, 109, 107574.
DOI	<a href="https://doi.org/10.1016/j.asoc.2021.107574">10.1016/j.asoc.2021.107574</a>
Publisher	Elsevier
Journal	Applied Soft Computing
Download date	2026-06-17 03:27:01
License	<a href="https://creativecommons.org/licenses/by-nc-nd/4.0/">https://creativecommons.org/licenses/by-nc-nd/4.0/</a>
Link to Item	<a href="http://hdl.handle.net/2436/624096">http://hdl.handle.net/2436/624096</a>

# Nonlinear-based Chaotic Harris Hawks Optimizer: Algorithm and Internet of Vehicles Application

Amin Abdollahi Dehkordi<sup>a</sup>, Ali Safaa Sadiq<sup>b</sup>, Seyedali Mirjalili<sup>c,d</sup>, Kayhan Zrar Ghafoor<sup>e</sup>

<sup>a</sup>Computer Engineering Faculty, Najafabad Branch, Islamic Azad University, Najafabad, Iran

<sup>b</sup>School of Mathematics and Computer Science, University of Wolverhampton, Wulfruna Street, Wolverhampton, WV1 1LY, UK

<sup>c</sup>Centre of Artificial Intelligence Research and Optimisation, Torrens University, Brisbane, Australia

<sup>d</sup>Yonsei Frontier Lab, Yonsei University, Seoul, Korea

<sup>e</sup>Department of Software Engineering, College of Engineering, Salahaddin University-Erbil

## Abstract

Harris Hawks Optimizer (HHO) is one of the many recent algorithms in the field of metaheuristics. The HHO algorithm mimics the cooperative behavior of Harris Hawks and their foraging behavior in nature called surprise pounce. HHO benefits from a small number of controlling parameters setting, simplicity of implementation, and a high level of exploration and exploitation. To alleviate the drawbacks of this algorithm, a modified version called Nonlinear based Chaotic Harris Hawks Optimization (NCHHO) is proposed in this paper. NCHHO uses chaotic and nonlinear control parameters to improve HHO's optimization performance. The main goal of using the chaotic maps in the proposed method is to improve the exploratory behavior of HHO. In addition, this paper introduces a nonlinear control parameter to adjust HHO's exploratory and exploitative behaviors. The proposed NCHHO algorithm shows an improved performance using a variety of chaotic maps that were implemented to identify the most effective one and tested on several well-known benchmark functions. The paper also considers solving an Internet of Vehicles (IoV) optimization problem that showcases the applicability of NCHHO in solving large-scale, real-world problems. The results demonstrate that the NCHHO algorithm is very competitive, and often superior, compared to the other algorithms. In particular, NCHHO provides 92% better results in average to solve the uni-modal and multi-modal functions with problem dimension sizes of  $D = 30$  and  $50$ , whereas, with respect to the higher dimension problem, our proposed algorithm shows 100% consistent improvement with  $D = 100$  and  $1000$  compared to other algorithms. In solving the IoV problem, the success rate was 62.5%, which is substantially better in comparison with the state-of-the-art algorithms. To this end, the proposed NCHHO algorithm in this paper demonstrates a promising method to be widely used by different applications, which brings benefits to industries and businesses in solving their optimization problems experienced daily, such as resource allocation, information retrieval, finding the optimal path for sending data over networks, path planning, and so many other applications.

**Keywords:** Optimization; Artificial intelligence; Harris hawks optimization algorithm; Chaos theory; Internet of Vehicles

## 1. Introduction

Over the past decade, challenging real-world optimization problems have been the center of attention in many areas, including but not limited to, information science-related problems, data analytics, engineering design, and wireless communication-related problems. Such optimization problems typically have common characteristics such as constraints, decision variables, objectives, etc. [1], [2]. Two common classes of optimization algorithms are used to solve such problems: conventional and modern algorithms. In the former class, there are usually mathematical optimization algorithms (mostly gradient-based), which suffer from local optima stagnation. In the latter class, however, metaheuristics are not gradient-based that benefit from a high chance of local optima avoidance, due to their stochastic behaviour [3],[4].

Simple implementation, robustness, and high efficiency are the superior characteristics of metaheuristic approaches. These properties assist metaheuristics to overcome the drawbacks of traditional approaches: premature convergence and local optima stagnation [5], [6]. As metaheuristic methods belong to the stochastic optimization family, they benefit from random operators to better avoid locally optimal solutions when solving optimization problems [7], [8]. Due to these distinct characteristics, metaheuristics are applicable to a wide range of problems in both science and industry. A common drawback for most metaheuristic algorithms, however, is that they often show resistance to user-defined parameter adjustments [9]. Moreover, metaheuristics may not always converge to global optimum solutions [10]. In addition, due to the use of a population of the solution, they tend to be computationally more expensive than conventional optimization algorithms.

Generally speaking, metaheuristic methods can be categorized into two classes: Evolutionary Algorithms (EA) and swarm intelligence (SI) techniques [11]. EA imitate the evolutionary mechanisms in nature. Some of the evolution-inspired techniques in this class are Evolutionary Strategy (ES) [12], Genetic Algorithms (GA) [13], Differential Evolution (DE) [14] and Biogeography-Based Optimization (BBO) algorithm [15]. However, to avoid being trapped in a local point of the solution in these methods, their parameters need to be optimally selected before the commencement of optimization. As such, a number of studies have been conducted to find the best set of parameters for such evolutionary algorithms.

SI methods, on the other hand, mimic the collective behavior of herds, schools, or flock in nature. To combat the difficulties of an optimization process, the relationship between the individuals in swarms is simulated in such methods. In nature, SI methods produce more than one random solution and improve them over the course of optimization process. Ant Colony Optimization (ACO) [16] and Particle Swarm Optimization (PSO) [17] are the two most popular algorithms that fall under this category. The ACO algorithm emulates ants' social behavior to determine the shortest path between the nest and a food source. The PSO algorithm simulates the birds' collective movement and hunting behavior. Other recent SI methods are: Salp Swarm Algorithm (SSA) [7], Cuckoo Search (CS) algorithm [18], Fruit Fly Optimization Algorithm (FOA) [19], Dolphin Echolocation (DE) [20], Grey Wolf Optimizer (GWO) [21], Bat Algorithm (BA) [22], Whale Optimization Algorithm (WOA) [23], Ant Lion Optimizer (ALO) [24] and Artificial Bee Colony (ABC) algorithm [25].

There is a common feature between these despite the differences in these algorithms; meta-heuristics divide the search process into two phases: exploration and exploitation. The exploration phase takes place when a meta-heuristic algorithm attempts to identify the best areas of a given search space. An algorithm should use its randomized operators to thoroughly explore diverse areas of the search space [9]. In contrast, the exploitation phase allows the optimizer to concentrate on the neighborhood that consists of better-quality solutions within the searching space. Due to conflicting nature of exploration and exploitation, a well-structured optimizer should be able to achieve a proper balance between the exploratory and exploitative tendencies. The probability of being stuck in local optima will be much higher in case of an improper balance between exploration and exploitation [26], [27].

Recently, a new metaheuristics technique, Harris Hawks Optimizer (HHO), has been introduced in [9]. This algorithm mimics Harris Hawks' hunting skills to catch their preys. The HHO is used to solve global optimization problems using these mechanisms. This algorithm consists of six steps of exploration and exploitation that make HHO capable of avoiding locally optimal solutions and boosting the convergence rate. We observed that its capability to exploit is more prominent compared to exploration [28].

As mentioned earlier, balancing the two phases of exploration and exploitation is one of the most important features in meta-heuristics that make these algorithms more capable of solving optimization problems. As a way to advance the performance of the basic HHO and improve its exploration phase, this paper modifies the HHO algorithm using different chaotic maps, which are utilized to change the random behavior of the basic HHO's parameters. Although the use of chaotic maps improves the HHO algorithm's exploration phase, it has the potential to disturb the tuning and the switchover between the exploration and exploitation phases. This challenging adaptation process will eventually

lead the algorithm to be trapped in local optimal points of solutions, thus hindering it from finding better competent solutions.

Therefore, the existence of a controlling parameter is crucial for balancing the algorithm's two stages. As another contribution of this work, a nonlinear parameter from Salp Swarm Algorithm (SSA) [6] is proposed as a control parameter operator for adjusting the exploration and exploitation phases. Accordingly, the proposed method is named as a Nonlinear-based Chaotic Harris Hawks Optimizer (NCHHO). The efficiency of the proposed NCHHO algorithm in finding the optimal solution is benchmarked in solving a wide range of well-known optimization test functions as well as in solving a real-world problem of finding the optimal network route on the Internet of Vehicles (IoV).

It should be noted that there are similar works in the literature on chaotic metaheuristics that will be extensively reviewed and discussed in the next section. However, not many research works targeted large-scale problems as well as IoV problems. We have tailored and modified the mechanism of HHO for large-scale problems, including IoV problems for the first time in the literature.

In the case of HHO, it is evident that this algorithm is not a suitable algorithm for large-scale problems. As a soft computing method, HHO requires improvements and modifications. Our improvements will be targeted to solve a very challenging case study in the area of IoV. Therefore, the following contributions are made for the first time in the literature of meta-heuristic and IoV.

- The use of chaotic map in HHO instead of random components to improve its exploration is the first contribution of this paper. We leverage chaotic maps to improve the exploratory behavior of HHO without increasing its computational cost, which was identified as one of the shortcomings of the existing chaotic meta-heuristics. While the use of chaotic values is successful in improving the algorithm's exploration process, it also creates a disruption between the exploration phase and the algorithm's exploitation, which is the motivation of the next contribution.
- To improve the accuracy and quality of selected best solutions in addition to improvising the exploitation ability of HHO, a nonlinear parameter is proposed for the first time in the literature for this algorithm.
- The last contribution is the use of NCHHO in solving one of the challenging optimization problems in the area of wireless communication for IoV as a seminal attempt to tackle this challenging problem using metaheuristics. This contribution will then facilitate a cost-effective, robust solution for the field of autonomous vehicles to improve data dissemination over a reliable optimal route.

Therefore, the novelty is on the modification of HHO in an attempt to solve large-scale problems including IoV problems.

The rest of the paper is organized as follows. Section 2 demonstrates the state-of-the-art in metaheuristics and their applications. This section also highlights the research gaps targeted in this work. In addition to defining different types of chaotic maps that will be hybridized and examined with HHO algorithm, Section 3 also briefs the readers with the main concepts and preliminaries of the basic HHO algorithm. In Section 4, we have demonstrated the detailed design of the proposed algorithm. Section 5 presents the comparative results and discussion to verify the performance and efficiency of the proposed NCHHO algorithm. Finally, Section 6 concludes the paper and highlights future works.

## 2. Literature Review

In the last few years, metaheuristics have established a trustworthy tie with a wide range of industries due to their strength in solving numerous real-world problems effectively. A substantial number of applications have adopted metaheuristic algorithms to solve problems with diverse difficulties, such

as single-objective, multi-objective, combinatorial, and constrained [29], [30]. As a way to brief the readers with the existing and most recent metaheuristic algorithms along with some real-world problems, this section presents some of these algorithms accordingly. Please note that we will focus on the IoV network traffic routing problem, as it has been selected to show case the applicability of the proposed NCHHO algorithm in solving such a highly dynamic problem with large dimensions.

We can observe from the literature that metaheuristics are widely adopted in solving problems with a nature of minimization, maximization, and mixed mode, with multi-constraints as well as feature selection problems [72]. While metaheuristic algorithms attempt to solve such kind of problems, an approximation is inherently used by applying a combination of incremental convergence approaches along with local and population-based search schemes. Such a strategy will facilitate the ability of local optima avoidance in metaheuristics. One of the main challenges that has been constantly argued about metaheuristic algorithms is their stochastic nature [31]. Despite being non-deterministic, metaheuristic algorithms can substantially save computational resources and make educated decisions to reduce the search space, whereas other exact type of algorithms struggle. Except for a given problem that could be solved by a deterministic algorithm within polynomial time that is equivalent to a non-deterministic algorithm  $P = NP$ , an exact method will not be able to solve an NP-hard optimization problem; worst case scenario will take polynomial time (with an exponential growth over the problem size). There are numerous research studies on investigating ways to bring exact method processing with maintained exponential time, yet this is very challenging to achieve. [32], [75], [76].

In spite of the fact that metaheuristics, on some occasions, could not provide very high quality solutions that meet 100% of the pre-defined objectives on a given optimization problem, they are considered as the most promising methods in solving real-life problems due to their time efficiency [73]. For instance, from the literature, we can observe that conventional metaheuristics, such as GA, PSO, GWO, etc., have demonstrated high potential in solving problems with complex behavior, clustering, and classification. This has motivated a lot of researchers to find new strategies or improve some of the existing algorithms to develop metaheuristic algorithms that are derived from nature phenomena such as Multi-Verse Optimizer (MVO) [50] and Volcano Eruption Optimizer (VEO) [33]; or inspired from natural creatures such as GWO, SSA, WOA; Hybrid multi-objective Discrete Artificial Bee Colony (HDABC) algorithm for the Blocking Lot-Streaming Flow Shop (BLSFS) scheduling problem [34] and many others.

It is worth mentioning that the well-known No Free Lunch (NFL) theorem [35] has logically proven that there is no metaheuristic algorithm that can produce the best results for all given optimization problems. In this context, several metaheuristic algorithms have been recently developed by researchers across the globe to find the optimal design that can be used to solve a wide range of optimization problems. For instance, Mafarja and Mirjalili [36] introduced two variants of the WOA algorithm: the roulette wheel and tournament selection variants. These newly introduced variants have replaced the conventional random operator that is used by WOA. Additionally, they have introduced the crossover and mutation operators as a way to improve the algorithm's performance. The limitation of their method is the degraded exploitation of the algorithm, especially when solving large-scale datasets.

On the other hand, in [37], the authors proposed the use of the chaotic search combined with WOA to overcome the problems of trapping within the local optima and leveraging the convergence speed, which eventually contributes to their showcase scenario of the feature selection problems. The limitation of their method is the lack of in-depth analysis of the impact of chaotic map on exploratory and exploitative behavior of WOA. As another attempt in [38], a filter algorithm utilizing Pearson correlation coefficient and correlation distance has been developed along with WOA. Their proposed technique intended to improve the algorithm in extracting the optimal set of features of a given optimization problem. The limitation of their method is the high computational cost of calculating the distance in each iteration. This is a real issue when solving real-world problems with computationally expensive objective functions. Hence, we could figure out that there are shortcomings within the

existing work, which are mainly high computational cost and degraded exploitation when solving large-scale problems.

As discussed above, HHO is a new algorithm that is proposed in [9] with a promising performance in solving optimization problems. However, this algorithm has a potential gap that needs to be revisited as a way to improve its performance. The main drawback of this algorithm is that it has a degraded exploitation when it comes to solving problems with high dimensions and dynamic behavior. Besides, in the HHO algorithm, the exploration is conducted based on normal distribution and random values. This nature of the population's initialization could lead to a non-strategic way of creating the initial population of solutions, which eventually impacts the computational time and quality of solutions (the details of this algorithm are discussed further in Section 3.1).

Chaos theory is a popular way of balancing exploration and exploitation in metaheuristics. In nonlinear dynamic systems, chaos is a stochastic phenomenon that is generated due to the fact that regularity, randomness, sensitivity to the initial condition, and periodicity are most likely experienced in the initialization phase of SI algorithms. [39]. Such chaos properties allow the algorithms to work with generic probability distributions at higher speeds than standard stochastic search and avoid being trapped in the optimal local stage [40], [41]. With the advantages of chaos theory and metaheuristic methods, in recent years, these two methods have shown promising potential to improve the performance of metaheuristics. Some examples are Chaotic Krill Herd [42], Fruit fly Optimization Algorithm (FOA) [43], Runner-Root Algorithm [44], Chaotic PSO [45], [46] Chaotic Gravitational Search Algorithm [47], and Chaotic WOA (CMWOA) [48][49]. Accordingly, the efficiency of metaheuristic methods was enhanced by utilizing this type of hybridization, which imposes the importance of chaos theory.

Recently, metaheuristic algorithms have been employed to solve optimization problems related to Vehicular Ad hoc Network (VANET) as well as Internet of Things (IoT) [74], particularly routing network traffic problem within the emerged era of IoV. For instance, in [50], García-Nieto et al. explored the use of several metaheuristic algorithms to find the optimal values of three defined configuration matrices like chunk size, total attempts, and retransmission time for the Vehicular Data Transfer Protocol (VDTP) for vehicular communications scenarios such as PSO, DE, GA, ES, and Simulated Annealing (SA). On the other hand, Optimization of Link State Routing (OLSR) protocol has been developed by the authors in [51], [52] for vehicular communications. Ghafor et al. [53] have developed a new metaheuristic algorithm named Laying Chicken Algorithm (LCA) to solve a defined discrete optimization problem in Software-defined Internet of Vehicles (SDIoV). LCA has been used in finding the optimal candidate route to be used in forwarding network traffic over smart vehicles in IoV. Although the LCA has shown promising results, there are still some limitations that could be further improved. For instance, the developed routing algorithm introduces extra overhead, which is due to the nature of LCA that inquires some time converging to the optimal solution using the heat transfer mechanism among solutions candidates.



**Fig. 1.** Heuristics versus meta-heuristics

In the area of metaheuristics, the algorithms are designed generic enough to solve a wide range of problems. They are not heuristic and problem-focused, but rather general-purpose. This is shown in Fig. 1. As can be seen, each of these two classes of optimization algorithms has its own pros and cons.

However, the literature evidently shows that metaheuristics have become extremely popular in both science and industry.

The main reason why metaheuristics have become popular is due to several reasons, of which the main one is its black-box nature. However, this does not mean that a metaheuristic is able to solve all optimization problems. The NFL theorem, as mentioned earlier, logically proved that there is no metaheuristic to solve all optimization problems. Therefore, we need to tailor a metaheuristics to solve targeted problems. This does not make them problem-focused as heuristics, but modifications and tunings are required to ensure high accuracy.

Table 1 summarizes and compares the most recent metaheuristic algorithms in the literature then benchmark them with the proposed NCHHO algorithm.

TABLE 1 Characteristic summary of popular algorithms

Algorithms/Features	Population-based?	Number of controlling parameters (pop size, iteration, etc)	Number of best solutions considered to update positions	Levy flight mechanism?	Considering Non-Linear Operator?	Considering Chaotic Operator?	Inspiration
SSA	✓	3	1	N	N	N	Swarm
MFO	✓	3	n	N	N	N	Swarm
GWO	✓	3	3	N	N	N	Swarm
PSO	✓	5	n+1	N	N	N	Swarm
DE	✓	4	0	N	N	N	Math
SA	✗	2	1	N	N	N	Physics
MVO	✓	4	n	N	N	N	Physics
HHO	✓	4	n	Y	N	N	Swarm
<b>NCHHO</b>	✓	<b>4</b>	<b>n</b>	<b>Y</b>	<b>Y</b>	<b>Y</b>	<b>Swarm</b>

The subsequent sections present the preliminaries and definitions of standard HHO and the proposed NCHHO algorithms. Also, we will introduce improvements to the HHO algorithm to alleviate its drawbacks when solving optimization problems, including solving a wide range of complex optimization functions and finding an optimal network traffic route for IoV applications.

### 3. Preliminaries and definition

This section introduces the preliminaries and the essential definitions of the HHO algorithm and the Chaotic Maps that will be used in developing the proposed NCHHO algorithm.

#### 3.1. Harris Hawks optimization (HHO)

The HHO algorithm was proposed in 2019 [9], which mimics some aspects of the Harris Hawks' social hunting mechanism. Harris's Hawks are medium-sized birds with a distinct dark brown color that usually live in semi-open desert lowlands. They can be found in urban regions too. They tend to hunt and migrate in groups. In finding the optimal solution to an optimization problem, the HHO uses three steps. These steps include exploration, shifting from exploration to exploitation, and exploitation [9].

##### 3.1.1. Exploration phase

During the search space exploration (selecting the appropriate prey (rabbit)) as described in Eq. (1), the HHO used two strategies to simulate the Harris Hawks' behaviors [9].

$$x_i(t+1) = \begin{cases} X_{rand}(t) - r_1|X_{rand}(t) - 2r_2X_i(t)| & q \geq 0.5 \\ (X^*(t) - X_m(t)) - r_3(lb + r_4(ub - lb)) & q \leq 0.5 \end{cases} \quad (1)$$

In Eq. (1),  $x_i(t+1)$  is the position vector of the hawks in the next iteration. Meanwhile,  $X^*(t)$  specifies the position of the prey,  $X_i(t)$  consists the current position vector of hawks.  $r_1, r_2, r_3, r_4$ , and  $q$  are random numbers generated at interval 0, 1 in each iteration.  $lb$  and  $ub$  are the upper and lower bounds of the variables. Randomly selected solutions from the current population of hawks and the average of the solutions (i.e., the current position of hawk) are placed in  $X_{rand}(t)$  and  $X_m(t)$ , respectively.  $X_m(t)$  is attained using Eq. (2) [9]:

$$X_m(t) = \frac{1}{N} \sum_{i=1}^N X_i(t) \quad (2)$$

where  $N$  refers to the total number of hawks.

### 3.1.2. Shifting from exploration to exploitation

According to the energy of the prey ( $E$ ) in HHO, the algorithm changes from exploration to exploitation phase. This concept has been defined as follows [9]:

$$E = 2E_0 \left(1 - \frac{t}{T}\right), \quad E_0 \in [-1, 1] \quad (3)$$

where  $E$  is the energy of the prey,  $T$  and  $t$  indicates the maximum number of iteration and the current iteration, respectively.  $E_0$  is the initial prey energy state that changes randomly within the range  $(-1, 1)$  at each iteration.

### 3.1.3. Exploitation phase

In this phase, by mimicking the prey's attacking strategy discussed in the previous process, Harris' hawks perform the surprise pounce. However, preys often try to evade dangerous situations. Suppose  $c$  is the chance of a prey to successfully escape from the hawks' siege ( $c < 0.5$ ) or to be trapped by them ( $c \geq 0.5$ ) before the surprise pounce. Hawks will use a hard or soft siege to hunt their prey, regardless of the different escape strategies performed by the prey. In other words, hawks encircle their prey in different ways, depending on the energy that remains with the prey [9].

Different patterns of chasing exist in real situations. In the HHO, four potential strategies are suggested to model the attacking stage based on common escaping behaviors of the prey and chasing patterns of the Harris' hawks [9]. These strategies include soft and hard besieges: a soft besiege with progressive rapid dives, and a hard besiege with progressive rapid dives.

The hawks will finally conduct a hard or soft besiege to hunt the prey [9]. In nature, by performing the surprise pounce, hawks get closer and closer to the targeted prey to reduce the risk of losing their hunt, and eventually catch the prey. As time passes, the prey will get exhausted and lose its energy. Then, the last step of hunting begins; the hawks will step up the process of besieging the prey for killing it. In the HHO algorithm,  $E$  parameter is used for simulating this hunting pattern and enabling the HHO to move between processes of soft and hard besieging. When  $|E| \geq 0.5$ , the hawks do a soft besiege and if  $|E| < 0.5$ , a hard besiege has occurred.

#### 3.1.3.1. Soft besiege

In this phase, the prey still has enough energy and tries to escape. It is softly surrounded by the Harris' hawks to exhaust the prey, after which the surprise pounce will be performed[9]. This behavior is implemented by the following rules:

$$x_i(t+1) = \Delta x_i(t) - E|J \times X^*(t) - x_i(t)|, \quad J = 2(1 - r_5) \quad (4)$$

$$\Delta x_i(t) = X^*(t) - x_i(t) \quad (5)$$

where  $J$  indicates the jump intensity of the prey (rabbit), which changes its value randomly in each iteration.  $r_5$  is a random number generated inside the range 0 to 1.

### 3.1.3.2. Hard besiege

When the prey has low escape energy, the Harris hawks hardly surround the prey for surprise pounce. This phase is called hard besieges in the HHO. In this phase, the solutions are updated using Eq. (6):

$$x_i(t+1) = X^*(t) - E|\Delta x_i(t)| \quad (6)$$

### 3.1.3.3. Soft besiege with progressive rapid dives

In the soft besiege with progressive rapid dives, the prey has enough energy to escape from Harris' hawk successfully, even before the surprise pounce; a soft besiege is shaped step by step [9]. This concept is performed using Eq. (7):

$$Y(t) = X^*(t) - E|J \times X^*(t) - x_i(t)| \quad (7)$$

The HHO uses levy flight ( $LF$ ) to simulate real zigzag movements of prey (especially rabbits) during the escape phase. So the levy flight is used to update Eq. (8) as [9]:

$$Z(t) = Y(t) + S \times LF(D) \quad (8)$$

In Eq. (8)  $D$  is a problem dimension,  $S$  is a random vector by size  $1 \times D$ , and  $LF$  is a levy flight function that is calculated using Eq. (9) [54] as follows:

$$LF(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}}, \sigma = \left( \frac{\Gamma(1 + \beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1 + \beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{\frac{1}{\beta}} \quad (9)$$

where  $u, v$  are random values generated in the range of 0 to 1, and  $\beta$  is set to 1.5. Therefore, the final step for updating the position of current solutions in this phase will be performed by Eq. (10) as follows [9]:

$$x_i(t+1) = \begin{cases} Y(t) & \text{if } F(Y(t)) < F(x_i(t)) \\ Z(t) & \text{if } F(Z(t)) < F(x_i(t)) \end{cases} \quad (10)$$

### 3.1.3.4. Hard besiege with progressive rapid dives

In comparison to the previous phase, the prey has low energy in this phase, and a strong besiege is designed to track and kill the prey before the surprise pounce[9]. So, the solutions are updated by Eq. (11) as follows [9]:

$$x_i(t+1) = \begin{cases} Y'(t) & \text{if } F(Y'(t)) < F(x_i(t)) \\ Z'(t) & \text{if } F(Z'(t)) < F(x_i(t)) \end{cases} \quad (11)$$

where  $Y$  and  $Z$  are calculated using new formulas in Eqs. (12) and (13) [9],

$$Y'(t) = X^*(t) - E|J \times X^*(t) - x_m(t)| \quad (12)$$

$$Z'(t) = Y'(t) + S \times LF(D) \quad (13)$$

## 3.2. Proposed chaotic maps for HHO

The development of random sequences with a long period and a high degree of accuracy are very important for the uncompromising simulation of complex phenomena, sampling, statistical evaluation, decision-making, and heuristic optimization in particular [55]. Chaos behavior could generally be a

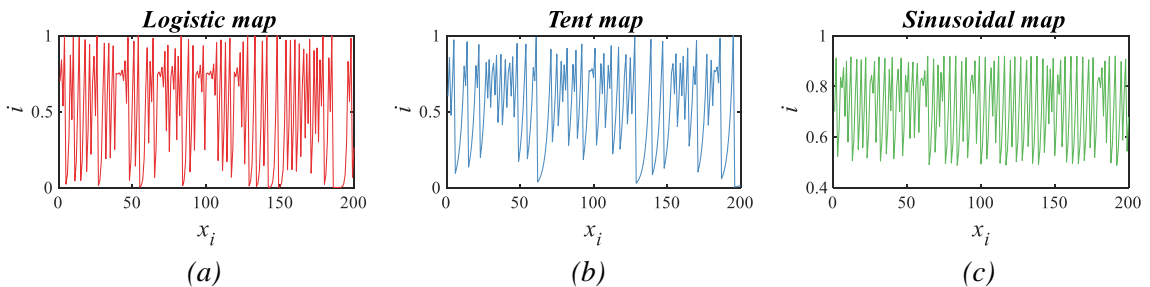
deterministic, random-like phenomenon that occurs in non-linear, dynamic (non-period), non-convergent, and bounded systems. Any number within the range [0-1] (or depending on the context of the chaotic map) can be selected as the initial value in these chaotic maps. However, it should be noted that the initial value could have significant impacts on the oscillation pattern of a few of the chaotic maps [56].

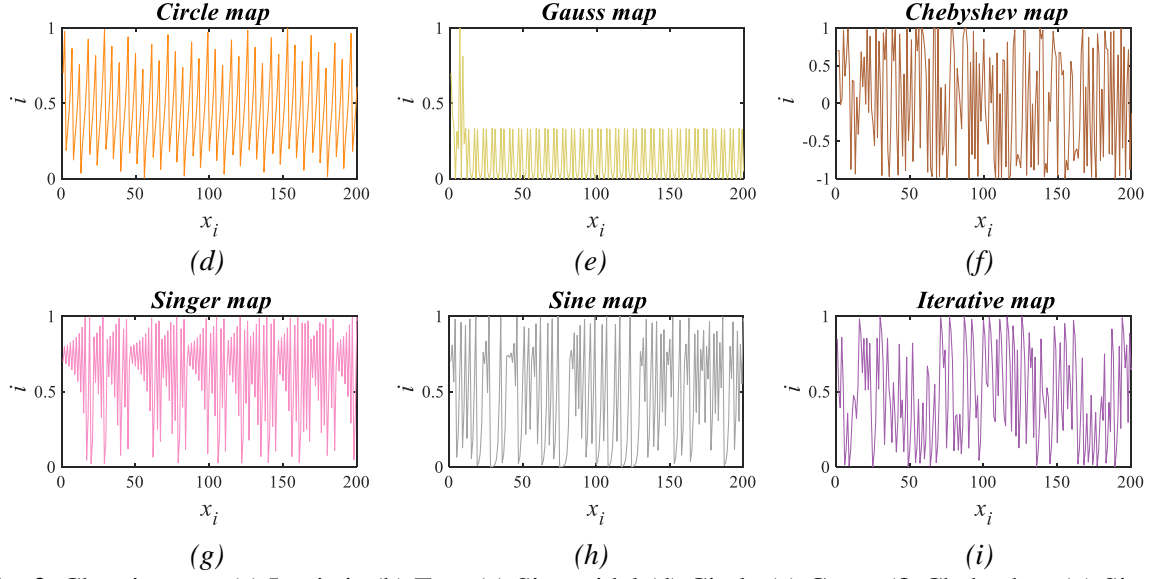
Many chaotic maps with various mathematical equations are used to propose chaotic optimization algorithms (COA) [42]. Due to the dynamic behavior of chaotic maps, during the last decade, researchers have been extensively adopted in the field of optimization. The reason is that these chaotic maps offer assistance to optimization algorithm in empowering the exploratory ability in the search space. In other words, exploration performed can be more systematic than stochastic, which is a probability-based search [57]. Furthermore, chaotic maps could effectively help optimization algorithms in avoiding locally optimal solutions and improving the convergence speed [28].

**Table 2** Chaotic maps and their equations used in improvising HHO's exploration phase.

No.	Map name	Map equation
1	Logistic map	$X_{n+1} = aX_n(1 - X_n), a = 4$
2	Tent map	$X_{n+1} = \begin{cases} 2X_n & X_n < 0.5 \\ 2(1 - X_n) & X_n \geq 0.5 \end{cases}$
3	sinusoidal map	$X_{n+1} = aX_n^2 \sin(\pi X_n)$
4	circle map	$X_{n+1} = X_n + b - \left(\frac{a}{2\pi}\right) \sin(2\pi X_n) \text{ mod}(1), a = 0.5 \text{ and } b = 0.2$
5	Gauss map	$X_{n+1} = \begin{cases} 0 & X_n = 0 \\ \frac{1}{X_n \text{ mod}(1)} & \text{otherwise} \end{cases}$
6	Chebyshev map	$X_{n+1} = \cos(a \cdot \cos^{-1} X_n)$
7	Singer map	$X_{n+1} = a(7.86X_n - 23.31X_n^2 + 28.75X_n^3 - 13.302875X_n^4), a = 1.07$
8	Sine	$X_{n+1} = \frac{a}{4} \sin(\pi X_n), a = 4$
9	Iterative	$X_{n+1} = \sin\left(\frac{a\pi}{X_n}\right), a = 0.7$

In order to improve the parameters of HHO algorithm, we use one-dimensional and other non-invertible maps to produce a set of chaotic values. Nine chaotic maps were used in experiments [57]–[59] as listed in Table 2. These equations were also visualized as a way to provide the readers with a more comprehensive understanding of their behavior. Fig. 2 demonstrates these nine selected chaotic maps that have been implemented and tested to show the impact on improving the exploration behavior of the proposed NCHHO algorithm. In each subfigure in Fig. 2, a chaotic map has been visualized out of the given related equation in Table 1. The presented behavior using 2-dimensional graphs was illustrated for Logistic, Tent, Sinusoidal, Circle, Gauss, Chebyshev, Singer, Sine and Iterative maps in Fig. 2(a)–(i) accordingly. The y-axis shows the values of the chaotic map while the x-axis presents the iteration numbers. As can be seen from each chaotic map's behavior, there are more added values for the covered regions and the frequency of movements. Hence, the use of each map will compensate each other in improvising the exploration behavior of an optimization algorithm. It is also important to mention that these maps are chosen because they have different behaviors in producing their chaotic values and have been shown to be successful in numerous studies in the literature [44], [60].





**Fig. 2.** Chaotic maps (a) Logistic (b) Tent (c) Sinusoidal (d) Circle (e) Gauss (f) Chebyshev (g) Singer (h) Sine (i) Iterative

#### 4. Nonlinear based Chaotic Harris Hawks Optimization

This section discusses the proposed algorithm's structure, which is named as Nonlinear based Chaotic Harris Hawks Optimization (NCHHO). The NCHHO's main goal is to advance the performance of the basic HHO algorithm in two ways; the first is to combine the HHO with the chaotic maps, while the second stage is the use of a non-linear control parameter as an adjustment operator between the exploration and exploitation phases. This combination in the basic HHO uses the behavior of chaotic maps that advance the strength of the algorithm to explore the search space while using the non-linear control parameter to improve the performance of HHO by adjusting the exploration and exploitation phases.

It is worth mentioning that the main purpose of using chaotic maps values is to deploy procedurally generated values rather than random values for NCHHO in the exploration phase. In addition, the use of chaos in the proposed method speeds up its performance compared to the original algorithm, which is a stochastic algorithm, and uses a standard probability distribution. In consideration of these advantages and in advance of the exploration phase in HHO, chaotic values have been utilized instead of random values in this process. Therefore, Eq. (1) is updated as follow:

$$x_i(t+1) = \begin{cases} X_{rand}(t) - cm_1 |X_{rand}(t) - 2cm_2 X_i(t)| & q \geq 0.5 \\ (X^*(t) - X_m(t)) - cm_3 (lb + cm_4 (ub - lb)) & q \leq 0.5 \end{cases} \quad (14)$$

where  $cm$  is a chaotic number based on the selected map generated in each iteration.

In Algorithm 1, the pseudo-code for the proposed NCHHO algorithm is stated.

---

##### **Algorithm 1** Pseudo-code of NCHHO algorithm

---

**Inputs:** The population (Harris' hawks) size  $N$  and maximum number of Algorithm iterations  $T$

**Outputs:** Position and fitness of the prey (rabbit) Initialize the random population  $X_{rand}$  ( $i = 1, 2, \dots, N$ )

**while** (stopping condition is not terminate) **do**  
  Calculate the hawks' fitness values  
  Set  $X_i$  as the location of prey (best location)

**for** (each hawk ( $X_{rand}$ )) **do**

    Update the initial energy  $E_0$  ( $E_0 = 2rand() - 1$ ) and jump strength  $J$  ( $J = 2(1 -$

---

---

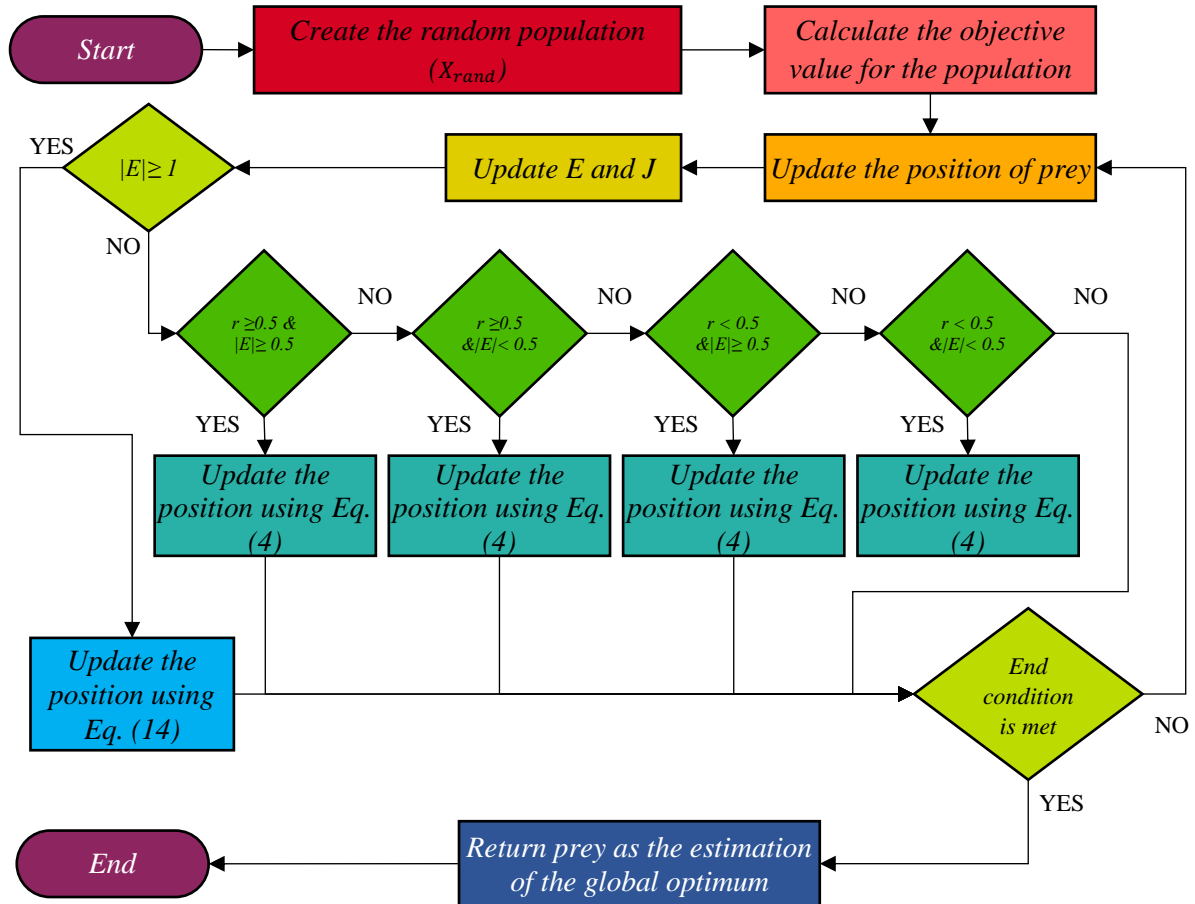
```

rand())
    Update Eq. (3)
►Exploration phase
    if ( $|E| \geq 1$ ) then
        Update the hawks location using Eq. (14)
►Exploitation phase
    if ( $|E| < 1$ ) then
►Soft besiege
        if ( $r \geq 0.5$  and  $|E| \geq 0.5$ ) then
            Update the hawks location using Eq. (4)
►Hard besiege
        else if ( $r \geq 0.5$  and  $|E| < 0.5$ ) then
            Update the hawks location using Eq. (6)
►Soft besiege with progressive rapid dives
        else if ( $r < 0.5$  and  $|E| \geq 0.5$ ) then
            Update the hawks location using Eq. (16)
►Hard besiege with progressive rapid dives
        else if ( $r < 0.5$  and  $|E| < 0.5$ ) then
            Update the hawks location using Eq. (19)
Return  $X_i$ 

```

---

To better see the algorithm of the proposed method, Fig. 3 is provided.



**Fig. 3.** The flowchart of the proposed method

The levels of exploration and exploitation are not only influenced by the variety of potential solutions, but are also controlled by different operators that monitor the intensity of search at various stages. [61]. Although the use of chaotic values improves the exploration phase of the HHO, it is very important to strike a balance between the exploration and exploitation phases of the

methods in order to achieve reasonable results. As a consequence, a nonlinear control parameter that was introduced in SSA [7], was implemented with a few modifications (the downward slope of the reduction of the value of this parameter has been changed according to the algorithm structure). A non-linear control parameter is used in our modified version of HHO algorithm to balance exploration and exploitation phases. Moreover, it makes the transition from exploration to exploitation in the proposed method to be executed smoothly. This parameter is defined as follow:

$$c = 2e^{-\left(\frac{8t}{T}\right)^2} \quad (15)$$

In Eq. (15),  $t$  is the current iteration and  $T$  is the total number of iterations.

We use the same equation in this work as well. As such, the process of soft besiege with progressive rapid dives in Eq. (10) has been modified in as follows:

$$x_i(t+1) = \begin{cases} Y(t) & \text{if } F(Y(t)) < F(x_i(t)) \\ Z(t) & \text{if } F(Z(t)) < F(x_i(t)) \end{cases} \quad (16)$$

$$Y(t) = cX^*(t) - E|J \times X^*(t) - x_i(t)| \quad (17)$$

$$Z(t) = cY(t) + S \times LF(D) \quad (18)$$

Moreover, in hard besiege with progressive rapid dives, Eq. (11) is modified using the following equations:

$$x_i(t+1) = \begin{cases} Y'(t) & \text{if } F(Y'(t)) < F(x_i(t)) \\ Z'(t) & \text{if } F(Z'(t)) < F(x_i(t)) \end{cases} \quad (19)$$

$$Y'(t) = cX^*(t) - E|J \times X^*(t) - x_m(t)| \quad (20)$$

$$Z'(t) = cY'(t) + S \times LF(D) \quad (21)$$

Note that the computational complexity of the NCHHO algorithm in theory is identical to that of HHO since chaotic maps and nonlinear equations are used instead of contact values for the main controlling parameters. Therefore, the improvement is done without bringing extra computational cost for the algorithm.

## 5. Results and discussion

This section first discusses the research methodology of this work and then presents the results. Just like other works on metaheuristics, the research methodology is quantitative. The proposed algorithm is tested on a wide range of test functions that will be introduced in following sub-section. Due to the stochastic nature of NCHHO, it is run 30 times and descriptive statistical measures are used to report its performance. The algorithm is compared with a number of conventional and recent optimization algorithms for results verification. Such comparisons are made using statistical measures and significance tests (95% confidence level) to ensure the results are not achieved by chance. In addition, the scalability and sensitivity analysis are also investigated to find out their impact on the performance of NCHHO. Finally, the problem definition along with the performance analysis for solving IoV routing optimization problem was demonstrated and discussed in this section.

It is important to highlight that the datasets used in this research are sets of benchmark mathematical functions, which have been used widely in the literature to evaluate the performance of optimization algorithms. Besides, we have also implemented a real-world problem (Internet-of-Vehicles) IoV routing problem, in which the dataset has been created based on the stated wireless channel characteristics model that are connectivity (Pc) and packet reception (Pr) of each route, taking into the consideration the constraint of route delay (D).

### 5.1. Benchmark functions' set details

Several experiments are carried out in this section to demonstrate the efficacy of the proposed method with theoretical remarks set out in the preceding sections. Based on the literature, a well-known set of specific benchmark functions is chosen, which includes 23 unimodal, multimodal and fixed-dimensional multimodal functions, and six composite functions which are commonly used to evaluate the performance of optimization algorithms[62], [63].

Unimodal functions (F1-F7), which have a single optimum solution, intentionally test the ability of the algorithm to exploit them (see Table 3). The second category covers multimodal functions (F8-F13) (Table 4) with more than one optimal solution. Local optimal solutions determine the exploratory behavior of the algorithm in these functions, while an algorithm needs to be able to search the space globally and avoid being trapped in local optimum to find the global optimum. Multi-modal fixed-dimensional functions (F14-F23), summarized in Table 5, are the other categories that are like multi-modal functions, but with low and fixed dimensions. Functions 24-29 include the composite benchmark functions introduced in CEC 2014 special session [64]. These functions have been built using shifted, rotated, expanded, and a combination of the most complex types of mathematical optimization problems presented in the literature [64]. These functions, along with their dimensions used in this study, are shown in Tables 3-6. **Table 3** Unimodal benchmark functions [71]–[73].

Functions	Dimensions	Range	$f_{min}$
$f_1(x) = \sum_{i=1}^n x_i^2$	30,100,500,1000	[-100,100]	0
$f_2(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	30,100,500,1000	[-10,10]	0
$f_3(x) = \sum_{i=1}^n \left( \sum_{j=1}^i x_j \right)^2$	30,100,500,1000	[-100,100]	0
$f_4(x) = \max_i \{ x_i , 1 \leq i \leq n\}$	30,100,500,1000	[-100,100]	0
$f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30,100,500,1000	[-30,30]	0
$f_6(x) = \sum_{i=1}^n ([x_i + 0.5])^2$	30,100,500,1000	[-100,100]	0
$f_7(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0,1]$	30,100,500,1000	[-1.28,1.28]	0

**Table 4** Multimodal benchmark functions [71]–[73].

Functions	Dimensions	Range	$f_{min}$
$f_8(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	30,100,500,1000	[-500,500]	-418.9829×n
$f_9(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30,100,500,1000	[-5.12,5.12]	0
$f_{10}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	30,100,500,1000	[-32,32]	0
$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30,100,500,1000	[-600,600]	0
$f_{12}(x) = \frac{\pi}{n} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\}$ $+ \sum_{i=1}^n u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4} u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 - a & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30,100,500,1000	[-50,50]	0
$f_{13}(x) = 0.1 \left\{ \sin^2(3\pi x_i) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \right\} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	30,100,500,1000	[-50,50]	0

**Table 4** Fixed-dimension multimodal benchmark functions [71]–[73].

Functions	Dimensions	Range	$f_{min}$
-----------	------------	-------	-----------

Functions	Dimensions	Range	$f_{min}$
$f_{14}(x) = \left( \frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right)^{-1}$	2	[-65,65]	1
$f_{15}(x) = \sum_{i=1}^{11} \left[ a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	[-5,5]	0.00030
$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[-5,5]	-1.0316
$f_{17}(x) = \left( x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6 \right)^2 + 10 \left( 1 - \frac{1}{8\pi} \right) \cos x_1 + 10$	2	[-5,5]	0.398
$f_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2] \times [18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2]$	2	[-2,2]	3
$f_{19}(x) = -\sum_{i=1}^4 c_i \exp \left( -\sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2 \right)$	3	[1,3]	-3.86
$f_{20}(x) = -\sum_{i=1}^4 c_i \exp \left( -\sum_{j=1}^6 a_{ij} (x_j - p_{ij})^2 \right)$	6	[0,1]	-3.32
$f_{21}(x) = -\sum_{i=1}^5 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.1532
$f_{22}(x) = -\sum_{i=1}^7 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.4028
$f_{23}(x) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.5363

**Table 6** Properties and summary of the composite CEC-BC-2014 functions (for the details of equations, please refer to [70]).

Functions	Dimensions	Range	$f_{min}$
CF24 =Composition Function of Schwefel's Function, Rotated Rastrigin's Function and Rotated HGBat Function	30	[-100,100]	2400
CF25 =Composition Function of Rotated Schwefel's Function, Rotated Rastrigin's Function and Rotated High Conditioned Elliptic Function	30	[-100,100]	2500
CF26 =Composition Function of Rotated Schwefel's Function, Rotated HappyCat Function, Rotated High Conditioned Elliptic Function, Rotated Weierstrass Function and Rotated Griewank's Function	30	[-100,100]	2600
CF27 =Composition Function of Rotated HGBat Function, Rotated Rastrigin's Function, Rotated Schwefel's Function, Rotated Weierstrass Function and Rotated High Conditioned Elliptic Function	30	[-100,100]	2700
CF28 =Composition Function of Rotated Expanded Griewank's plus Rosenbrock's Function, Rotated HappyCat Function, Rotated Schwefel's Function, Rotated Expanded Scaffer's and Rotated High Conditioned Elliptic Function	30	[-100,100]	2800
CF29 =Composition Function of Hybrid Function 1, Hybrid Function 2, Hybrid Function 3	30	[-100,100]	2900

## 5.2. Experimental Setup

We have compared the performance of the proposed NCHHO algorithm with a number of well-known algorithms in the literature, including DE [14], PSO [17], WOA [23], SSA [7], MVO[68], MFO[69], GWO [21] and the original HHO[9] to verify the efficiency and validate the results. The parameter settings of these algorithms are same as the original work (see Table 7). The proposed method and the other benchmarked algorithms were implemented in Matlab R2018a installed on a computer with a Windows 8.1 64-bit and 6 GB RAM. Moreover, all optimization algorithms' populations and iterations are set to 100 and 500 respectively.

**Table 7** The Algorithms parameter settings.

Algorithm	Parameter	Value
HHO	Rabbit Energy	[2 0]
	$c$ (Nonlinear control parameter)	[2 0]
NCHHO	Sine chaotic map( $X_{n+1} = \frac{a}{4} \sin(\pi X_n)$ , $a = 4$ , $X_n = 0.7$ )	[1 0]
	Topology fully connected Inertia factor	0.5
PSO	$c_1$	2
	$c_2$	2
DE	Scaling factor	0.5
	Crossover probability	0.2
MVO	maximum of Wormhole Existence Probability	1
	minimum of Wormhole Existence Probability	0.2
MFO	Convergence constant $a$	[-1 -2]
SSA	$c_1$ (balancing parameter of exploration and exploitation)	[1 0]

### 5.3. Performance Analysis of Non-linear Chaotic maps effects on NCHHO

In order to evaluate the performance of the proposed NCHHO algorithm and benchmark with the other state-of-the-art optimization algorithms, the listed fitness functions in Tables 2-5 were implemented, and the performance was measured for each algorithm. We started the evaluation by first finding out the best chaotic map that could achieve the highest performance with each fitness function of the set of F1-13. Table 7 lists the best average fitness function value calculated for each chaotic map and for each benchmark function over 30 runs. The results in Table 7 show that the NCHHO with the Sine map provides the best global solutions with five functions and has shown competitive results with the rest of them.

Therefore, the Sine map was chosen as the best map for NCHHO algorithm and was used in developing our algorithm and compare it with other optimization algorithms. The value of 0.7 is used as an initial value for all chaotic maps. In addition, all these maps are normalized in the range [0-1], to be used instead of random generators in HHO. To evaluate the results of all optimization algorithms and to specify whether the results are considerably different from each other, a Wilcoxon rank-sum test is used [70].

**Table 7** Results of NCHHO with different chaotic maps.

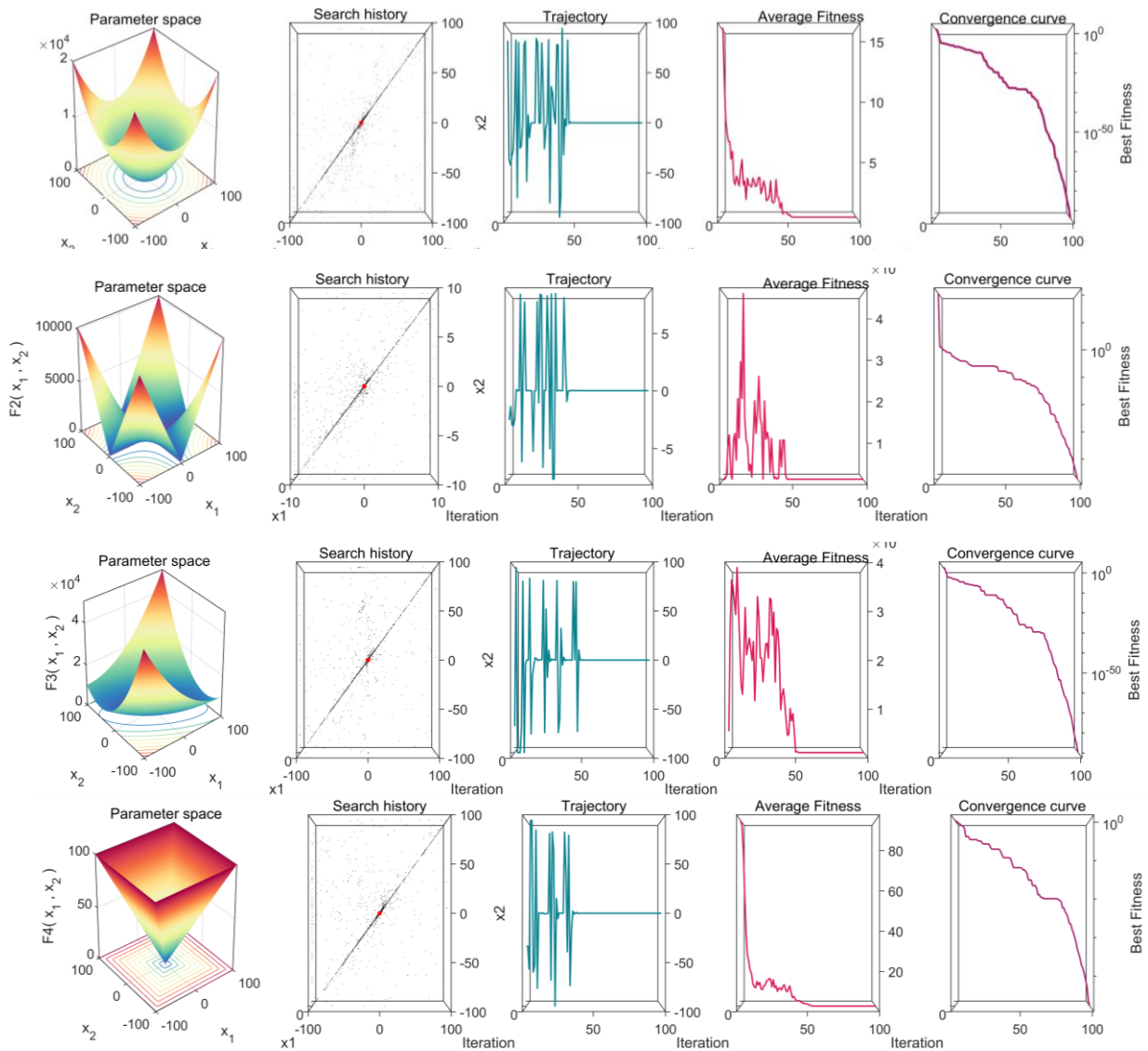
Func.	Chebyshev	Circle	Gauss-mouse	Iterative	Logistic	Sine	Singer	Sinusoidal	Tent
F1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F2	1.20E-227	1.82E-227	4.08E-228	3.12E-224	1.00E-223	<b>2.45E-229</b>	7.33E-225	1.85E-225	2.72E-225
F3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F4	9.28E-222	2.32E-228	<b>2.94E-230</b>	4.45E-227	6.04E-227	1.05E-225	7.54E-226	2.84E-226	2.26E-222
F5	<b>4.27E-03</b>	2.90E-02	6.37E-03	1.12E-02	1.21E-02	6.00E-03	1.81E-02	6.22E-03	1.51E-02
F6	3.11E-05	3.74E-05	4.70E-05	3.60E-05	3.29E-05	<b>1.35E-05</b>	5.33E-05	3.89E-05	4.29E-05
F7	1.20E-04	1.06E-04	1.04E-04	1.05E-04	1.42E-04	<b>8.08E-05</b>	1.18E-04	1.30E-04	1.28E-04
F8	-1.26E+04	-1.26E+04	-1.26E+04	-1.26E+04	-1.26E+04	-1.26E+04	-1.26E+04	-1.26E+04	-1.26E+04
F9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F10	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F12	6.58E-06	4.35E-06	5.83E-06	3.68E-06	2.29E-06	<b>2.10E-06</b>	3.18E-06	2.77E-06	5.19E-06
F13	4.00E-05	6.48E-05	3.60E-05	7.94E-05	4.27E-05	<b>2.01E-05</b>	5.96E-05	2.67E-05	3.87E-05
<b>Best</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>5</b>	<b>0</b>	<b>0</b>	<b>0</b>

### 5.4. Visualization of results

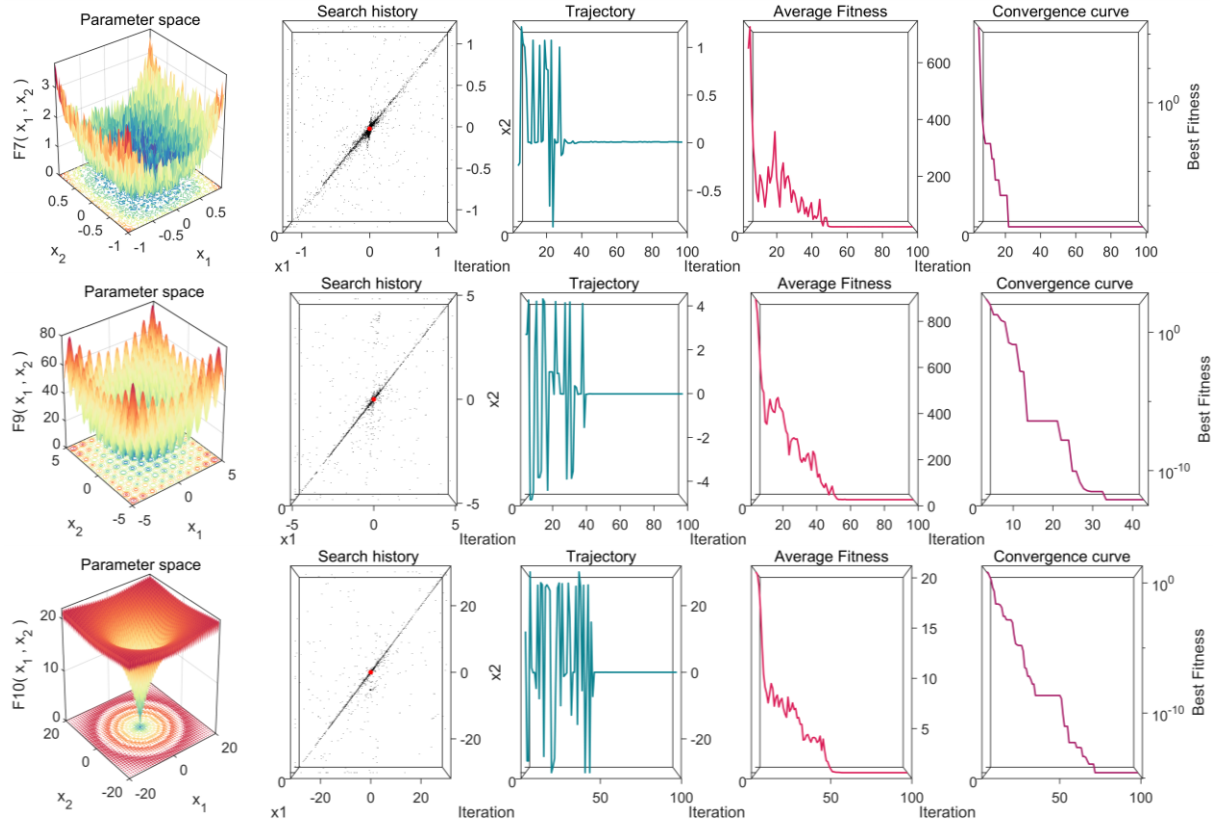
In this section, the results of the qualitative analysis of the proposed method when solving unimodal and multimodal functions are presented and analyzed. Qualitative analysis intuitively analyzes positional changes as well as fitness changes during the hawks' hunting process. Figs. 4-6 demonstrate four indicators: search history, the trajectory of the first hawk, average fitness, and convergence curve. The trajectory subplot tracks how the first hawk's first variable changes all through the optimization process. The average fitness curve shows the changes in the average fitness of population during optimization. Throughout the iteration process, the convergence curve indicates that the NCHHO has obtained the optimum fitness value with relatively smooth convergence towards the end of the course of iterations.

It can be seen in the subplot of the search history in Figs. 4-6 that the NCHHO shows a good coverage of the search space while focusing on promising regions. The trajectory diagram shows that the solutions in NCHHO have faced sudden changes in the initial stages that gradually taper off in the final steps. This reassures that the NCHHO ultimately converges to a position and exploits the region of interest [71]. The trajectories in Figs. 4-6 show a primary exploratory behavior of the algorithm due to sudden movements. The rapid amplitude in the initial iterations (covering even the 50% of the exploration space), and the slight amplitude in the later iterations can guarantee the rapid convergence of NCHHO and an effective search close to the global optimal [71], and also facilitates NCHHO's shift from exploratory to exploitative patterns.

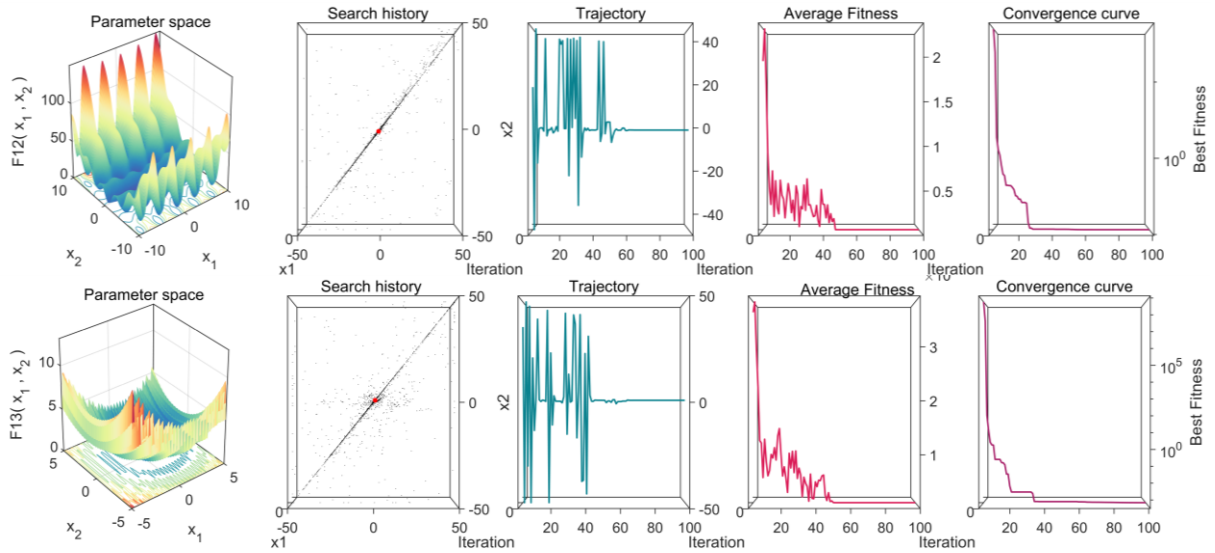
Moreover, this analysis can show NCHHO's attitude for exploration. Inspecting the average fitness curve shown in Figs. 4-6, the changes of the NCHHO fitness during the iterations has proved the positive impact of using chaotic maps. While the average fitness of NCHHO is deviated in some cases, the curve shows gradual descent, which is an indication of improving the overall quality of the population.



**Fig. 4.** Qualitative results for unimodal F1, F2, F3, and F4 benchmark functions.



**Fig. 5.** Qualitative results for F7, F9, and F10 benchmark functions.



**Fig. 6.** Qualitative results for F12 and F13 benchmark functions.

#### 5.4. Scalability Analysis

In this section, the scalability of the proposed NCHHO algorithm has been investigated and analyzed accordingly. The scalability evaluation is used in this section to examine the effect of different dimensions on NCHHO's performance. It also demonstrates how a metaheuristic can retain its higher-dimensional search advantages [9]. The stochastic nature of metaheuristics causes the results of a single run to be unreliable; therefore, all algorithms have been checked 30 times and statistical

measurements; mean, and standard deviations are collected and listed in Tables 8-12. For the first two groups of functions (F1-F13), the dimensions of 30,100,500 and 1000 are considered for the experiments. The obtained mean (AVG) and standard deviation (STD) results of all optimization algorithms for 30 runs with 500 iterations were collected and compared for each dimension.

**Table 9** Results of benchmark functions (F1–F13), with 30 dimensions.

Func.		WOA	SSA	MFO	GWO	PSO	DE	MVO	HHO	NCHHO
F1	Ave	1.42E-73	1.80E-07	3.68E+03	4.87E-41	2.36E-07	3.42E+03	5.74E-01	2.11E-99	<b>0.00E+00</b>
	Std	7.24E-73	1.84E-07	6.69E+03	5.50E-41	3.06E-07	9.96E+03	1.51E-01	9.11E-99	0.00E+00
F2	Ave	1.51E-49	1.58E+00	3.15E+01	4.78E-24	3.99E-04	1.99E+06	4.39E+00	8.43E-51	<b>2.61E-221</b>
	Std	8.19E-49	1.40E+00	2.12E+01	3.57E-24	3.98E-04	2.12E+07	2.07E+01	2.89E-50	0.00E+00
F3	Ave	4.29E+04	1.95E+03	2.15E+04	1.23E-11	1.91E+01	4.06E+04	6.42E+01	7.94E-74	<b>0.00E+00</b>
	Std	1.27E+04	1.34E+03	1.27E+04	5.32E-11	7.67E+00	1.11E+04	2.22E+01	4.31E-73	0.00E+00
F4	Ave	5.60E+01	1.05E+01	6.98E+01	1.88E-10	5.53E-01	3.61E+01	1.17E+00	1.55E-49	<b>4.25E-223</b>
	Std	2.60E+01	3.14E+00	9.05E+00	2.03E-10	1.14E-01	1.91E+01	3.57E-01	8.05E-49	0.00E+00
F5	Ave	2.79E+01	4.93E+02	8.01E+06	2.66E+01	5.97E+01	7.35E+06	3.53E+02	1.37E-02	<b>6.69E-03</b>
	Std	5.30E-01	1.09E+03	2.44E+07	8.70E-01	4.28E+01	2.71E+07	6.10E+02	2.62E-02	8.43E-03
F6	Ave	4.61E-01	1.53E-07	1.02E+03	2.27E-01	2.49E-07	3.51E+03	6.06E-01	2.63E-04	<b>4.84E-05</b>
	Std	2.68E-01	2.05E-07	3.08E+03	2.55E-01	3.88E-07	1.00E+04	1.74E-01	4.72E-04	1.04E-04
F7	Ave	2.54E-03	1.86E-01	-1.28E+00	5.69E-04	7.78E-02	4.05E+00	2.06E-02	1.55E-04	<b>1.40E-04</b>
	Std	2.21E-03	8.25E-02	1.03E+01	3.76E-04	2.94E-02	1.53E+01	6.97E-03	1.56E-04	1.26E-04
F8	Ave	-9.89E+03	-7.56E+03	-8.49E+03	-6.25E+03	-6.59E+03	-7.47E+03	-7.87E+03	-1.25E+04	<b>-1.26E+04</b>
	Std	1.69E+03	7.55E+02	8.70E+02	9.31E+02	7.25E+02	1.19E+03	5.21E+02	3.18E+02	2.27E+00
F9	Ave	-5.12E+00	5.33E+01	1.66E+02	8.60E-01	4.28E+01	1.38E+02	1.20E+02	<b>0.00E+00</b>	<b>0.00E+00</b>
	Std	0.00E+00	2.06E+01	2.77E+01	3.04E+00	9.33E+00	6.29E+01	3.17E+01	0.00E+00	0.00E+00
F10	Ave	3.97E-15	2.49E+00	1.39E+01	2.75E-14	2.62E-04	4.77E+00	1.21E+00	<b>8.88E-16</b>	<b>8.88E-16</b>
	Std	2.59E-15	7.87E-01	7.04E+00	4.35E-15	1.77E-04	6.24E+00	5.76E-01	0.00E+00	0.00E+00
F11	Ave	4.07E-03	1.86E-02	2.19E+01	2.31E-03	8.87E-03	3.00E+01	6.87E-01	<b>0.00E+00</b>	<b>0.00E+00</b>
	Std	2.23E-02	1.26E-02	3.86E+01	4.92E-03	9.00E-03	8.29E+01	8.78E-02	0.00E+00	0.00E+00
F12	Ave	1.39E-01	6.91E+00	1.04E+01	1.37E-02	3.46E-03	1.51E+07	1.47E+00	<b>5.26E-06</b>	5.84E-06
	Std	6.23E-01	3.75E+00	7.06E+00	1.04E-02	1.89E-02	6.54E+07	1.31E+00	5.89E-06	1.04E-05
F13	Ave	5.33E-01	1.58E+01	1.37E+07	2.10E-01	1.90E-03	2.53E+07	8.72E-02	1.03E-04	<b>3.29E-05</b>
	Std	2.65E-01	1.59E+01	7.49E+07	1.53E-01	4.15E-03	1.06E+08	3.68E-02	1.92E-04	6.33E-05

Table 9 indicates that the proposed method shows the best results. When dealing with 30-dimensional functions, NCHHO's results are significantly better than the original HHO and other algorithms. This is due to the fact that the proposed algorithm provides superior efficiency when it has been tuned with chaos and nonlinear control parameters. This has been proven by the obtained p-values in Table 15, which shows that the existing differences in results are statistically significant in all cases.

When dealing with 100-dimensional space-search functions, NCHHO can reliably outperform other techniques. The results in Table 10 show the superiority of the NCHHO over HHO and other techniques, in approximately all runs. In Table 11, it can be evidently observed that in most of the cases with 500 dimensions, the NCHHO has obtained better results in terms of AVG and STD than other optimizers, especially compared with the standard HHO. As per p-values in Table 15, it is evident that NCHHO shows significantly better performance than other optimizers with all tested cases. Results obtained by the proposed algorithm have significantly improved in comparison with those obtained by the other methods. In either case, the other methods were capable of achieving the results that were obtained by the proposed NCHHO algorithm. In other words, these results show that the use of chaotic map values and nonlinear control parameters seems to provide a positive effect on the performance of the proposed method.

As was observed in lower dimensions, we have applied high-dimensional optimization problems as a way to testify the performance of the proposed algorithm. Table 12 reveals that the NCHHO algorithm provides surprisingly superior performance compared to WOA, PSO, DE, SSA, MVO, GWO, MFO, and HHO. Wilcoxon's rank-sum test results are listed in Table 15 and demonstrate that NCHHO has the ability to outperform other methods with a great deal of difference. It is important to mention that NCHHO has not only reached the lower global optimum in almost all cases with any dimension than other techniques, but also has great performance as compared to HHO, in dealing with these fitness functions.

**Table 10** Results of benchmark functions (F1–F13), with 100 dimensions.

Func.		WOA	SSA	MFO	GWO	PSO	DE	MVO	HHO	NCHHO
F1	Ave	5.51E-71	1.41E+03	6.33E+04	1.78E-18	4.75E+00	5.31E+04	8.08E+01	9.05E-97	<b>0.00E+00</b>
	Std	2.97E-70	3.05E+02	1.65E+04	1.35E-18	1.58E+00	6.38E+04	1.34E+01	3.35E-96	0.00E+00
F2	Ave	7.19E-50	4.82E+01	2.44E+02	2.18E-11	1.44E+01	3.55E+42	1.85E+24	6.35E-49	<b>5.74E-225</b>
	Std	1.82E-49	5.68E+00	3.18E+01	6.62E-12	4.31E+00	5.47E+43	1.01E+25	3.47E-48	0.00E+00
F3	Ave	1.02E+06	5.42E+04	2.24E+05	1.73E+01	8.84E+03	4.58E+05	4.86E+04	1.80E-42	<b>0.00E+00</b>
	Std	3.16E+05	2.91E+04	4.47E+04	3.43E+01	1.87E+03	8.82E+04	6.55E+03	9.84E-42	0.00E+00
F4	Ave	7.85E+01	2.86E+01	9.28E+01	2.69E-02	8.10E+00	9.12E+01	5.06E+01	1.48E-48	<b>4.23E-225</b>
	Std	2.05E+01	4.02E+00	2.85E+00	5.85E-02	1.32E+00	2.17E+00	4.86E+00	6.80E-48	0.00E+00
F5	Ave	9.81E+01	1.77E+05	8.01E+06	9.71E+01	3.77E+03	1.81E+08	4.37E+03	9.75E-02	<b>4.42E-02</b>
	Std	2.69E-01	8.00E+04	2.44E+07	9.52E-01	1.62E+03	2.74E+08	4.82E+03	1.32E-01	9.95E-02
F6	Ave	3.89E+00	1.52E+03	1.84E+08	6.14E+00	4.36E+00	5.16E+04	7.73E+01	3.84E-04	<b>5.75E-05</b>
	Std	1.20E+00	3.44E+02	7.39E+07	8.62E-01	1.56E+00	6.18E+04	1.12E+01	6.46E-04	1.16E-04
F7	Ave	4.30E+00	2.94E+00	2.52E+02	2.24E-03	1.32E+03	2.29E+02	3.50E-01	2.22E-04	<b>1.93E-04</b>
	Std	1.32E+00	6.97E-01	1.05E+02	8.93E-04	2.55E+02	3.42E+02	7.45E-02	2.58E-04	2.38E-04
F8	Ave	-3.59E+04	-2.20E+04	-2.29E+04	-1.70E+04	-1.67E+04	-1.40E+04	-2.36E+04	-4.19E+04	<b>-4.19E+04</b>
	Std	6.22E+03	1.66E+03	2.40E+03	2.51E+03	3.71E+03	2.35E+03	1.36E+03	8.83E+00	2.51E+00
F9	Ave	1.14E-14	2.41E+02	8.45E+02	3.78E+00	4.46E+02	9.90E+02	6.46E+02	<b>0.00E+00</b>	<b>0.00E+00</b>
	Std	4.58E-14	3.51E+01	6.33E+01	5.24E+00	5.29E+01	1.88E+02	6.57E+01	0.00E+00	0.00E+00
F10	Ave	3.97E-15	1.04E+01	1.98E+01	1.61E-10	2.65E+00	1.55E+01	5.40E+00	<b>8.88E-16</b>	<b>8.88E-16</b>
	Std	2.23E-15	1.30E+00	1.64E-01	5.39E-11	3.09E-01	3.87E+00	4.04E+00	0.00E+00	0.00E+00
F11	Ave	9.79E-03	1.37E+01	5.20E+02	4.45E-04	8.26E-02	4.75E+02	1.72E+00	<b>0.00E+00</b>	<b>0.00E+00</b>
	Std	5.36E-02	4.25E+00	1.15E+02	2.44E-03	2.06E-02	5.81E+02	1.10E-01	0.00E+00	0.00E+00
F12	Ave	4.81E-02	3.40E+01	2.76E+08	1.53E-01	1.29E+00	4.03E+08	1.41E+01	2.38E-06	<b>1.45E-06</b>
	Std	2.14E-02	1.09E+01	2.07E+08	5.93E-02	7.67E-01	6.28E+08	3.96E+00	2.80E-06	2.22E-06
F13	Ave	3.08E+00	6.43E+03	5.97E+08	4.92E+00	1.16E+01	7.62E+08	1.24E+02	8.51E-05	<b>5.29E-05</b>
	Std	8.13E-01	8.15E+03	2.52E+08	4.14E-01	5.93E+00	1.15E+09	2.98E+01	1.14E-04	7.12E-05

**Table 11** Results of benchmark functions (F1–F13), with 500 dimensions.

Func.		WOA	SSA	MFO	GWO	PSO	DE	MVO	HHO	CHHO
F1	Ave	5.70E-69	9.61E+04	1.16E+06	1.09E-05	2.91E+03	1.21E+06	5.93E+04	5.67E-95	<b>0.00E+00</b>
	Std	2.77E-68	6.05E+03	3.43E+04	3.13E-06	1.59E+02	1.50E+05	3.60E+03	2.44E-94	0.00E+00
F2	Ave	6.72E-49	5.27E+02	7.99E+121	6.14E-04	1.99E+13	6.30E+257	4.64E+206	3.48E-50	<b>1.50E-225</b>
	Std	1.93E-48	1.57E+01	4.38E+122	9.24E-05	1.09E+14	2.17E+10	1.64E+36	1.01E-49	0.00E+00
F3	Ave	3.26E+07	1.45E+06	4.79E+06	1.55E+05	3.68E+05	1.10E+07	1.70E+06	2.33E-43	<b>0.00E+00</b>
	Std	1.08E+07	6.08E+05	9.95E+05	5.16E+04	6.71E+04	1.20E+06	1.18E+05	1.06E-42	0.00E+00
F4	Ave	8.16E+01	3.96E+01	9.89E+01	5.34E+01	2.39E+01	9.87E+01	9.36E+01	2.73E-49	<b>7.50E-224</b>
	Std	2.02E+01	2.24E+00	3.00E-01	5.85E+00	1.06E+00	6.12E-13	1.47E+00	1.19E-48	0.00E+00
F5	Ave	4.96E+02	3.77E+07	4.99E+09	4.97E+02	1.22E+07	6.50E+09	5.45E+07	1.83E-01	<b>8.07E-02</b>
	Std	4.73E-01	4.83E+06	1.92E+08	3.39E-01	1.33E+06	5.43E+08	9.62E+06	2.55E-01	1.24E-01
F6	Ave	3.25E+01	9.35E+04	1.16E+06	8.06E+01	2.86E+03	1.24E+06	5.79E+04	2.31E-03	<b>4.54E-04</b>
	Std	7.59E+00	7.15E+03	3.09E+04	2.10E+00	1.86E+02	1.38E+05	3.75E+03	4.69E-03	7.15E-04
F7	Ave	3.43E-03	2.78E+02	3.88E+04	1.49E-02	5.45E+04	4.46E+04	4.03E+02	1.63E-04	<b>1.44E-04</b>
	Std	3.29E-03	4.23E+01	2.51E+03	3.50E-03	2.04E+03	7.45E+03	5.49E+01	1.25E-04	1.30E-04
F8	Ave	-1.75E+05	-6.12E+04	-6.22E+04	-6.19E+04	-4.97E+04	-3.08E+04	-8.08E+04	-2.09E+05	<b>-2.09E+05</b>
	Std	3.01E+04	5.37E+03	5.98E+03	1.30E+04	1.01E+04	4.98E+03	2.49E+03	4.70E+03	2.01E+01
F9	Ave	<b>0.00E+00</b>	3.17E+03	6.97E+03	4.03E+01	5.80E+03	7.84E+03	6.05E+03	<b>0.00E+00</b>	<b>0.00E+00</b>
	Std	0.00E+00	1.10E+02	1.59E+02	1.46E+01	2.31E+02	2.88E+02	1.83E+02	0.00E+00	0.00E+00
F10	Ave	4.56E-15	1.42E+01	2.04E+01	1.59E-04	9.74E+00	2.09E+01	2.08E+01	<b>8.88E-16</b>	<b>8.88E-16</b>
	Std	2.72E-15	2.33E-01	1.13E-01	2.55E-05	2.77E-01	5.52E-02	5.79E-02	0.00E+00	0.00E+00
F11	Ave	3.70E-18	8.36E+02	1.04E+04	2.97E-03	2.93E+00	1.10E+04	5.36E+02	<b>0.00E+00</b>	<b>0.00E+00</b>
	Std	2.03E-17	4.35E+01	3.46E+02	1.14E-02	1.83E-01	1.48E+03	3.02E+01	0.00E+00	0.00E+00
F12	Ave	9.64E-02	1.25E+06	1.21E+10	6.37E-01	3.07E+04	1.61E+10	3.43E+07	1.85E-06	<b>9.44E-07</b>
	Std	4.91E-02	6.42E+05	8.21E+08	2.91E-02	1.49E+04	5.79E+08	1.18E+07	1.76E-06	1.86E-06
F13	Ave	1.80E+01	3.67E+07	2.25E+10	4.55E+01	8.24E+05	2.86E+10	1.33E+08	7.28E-04	<b>1.67E-04</b>
	Std	6.14E+00	1.03E+07	1.09E+09	7.89E-01	2.39E+05	1.69E+09	3.12E+07	1.85E-03	3.22E-04

The results in Table 13 have shown that NCHHO has very competitive results in dealing with F14–F23 cases. However, the best global results for F14–F23 can also be achieved with other approaches. From the results obtained, it can be deduced that whenever we have a good equilibrium between the two phases of exploration and exploitation in metaheuristic algorithms, such algorithms will perform well in handling optimization problems. The NCHHO algorithm benefits from the advantages of chaotic maps to boost the exploration phase. In addition, utilizing a nonlinear control parameter to the proposed method establishes a balance between the two phases of exploration and exploitation. Based on the results achieved by the proposed method compared to other techniques, the strategy used in the NCHHO algorithm has been shown to be efficient.

**Table 12** Results of benchmark functions (F1–F13), with 1000 dimensions.

Func.		WOA	SSA	MFO	GWO	PSO	DE	MVO	HHO	NCHHO
F1	Ave	6.92E-70	2.37E+05	2.72E+06	7.12E-03	2.24E+04	3.03E+06	5.57E+05	3.52E-94	<b>0.00E+00</b>
	Std	2.71E-69	8.54E+03	5.87E+04	1.73E-03	9.08E+02	6.38E+04	2.19E+04	1.91E-93	0.00E+00
F2	Ave	9.30E-47	1.19E+03	2.68E+10	8.74E-02	1.26E+03	4.58E+00	1.73E+289	1.34E-48	<b>1.40E-223</b>
	Std	3.76E-46	3.17E+01	1.54E+09	4.22E-02	5.02E+01	2.83E-01	5.73E+249	3.16E-48	0.00E+00
F3	Ave	1.23E+08	6.16E+06	1.93E+07	8.35E+05	1.58E+06	4.35E+07	6.70E+06	1.71E-24	<b>0.00E+00</b>
	Std	3.77E+07	2.43E+06	3.13E+06	2.33E+05	3.24E+05	9.41E+06	5.00E+05	9.32E-24	0.00E+00
F4	Ave	7.66E+01	4.43E+01	9.95E+01	7.26E+01	2.92E+01	9.96E+01	9.76E+01	6.58E-46	<b>5.42E-225</b>
	Std	2.51E+01	2.52E+00	1.95E-01	4.49E+00	9.14E-01	5.83E-13	6.58E-01	3.60E-45	0.00E+00
F5	Ave	9.94E+02	1.21E+08	1.26E+10	9.98E+02	1.56E+08	1.48E+10	1.33E+09	3.80E-01	<b>2.02E-01</b>
	Std	1.02E+00	1.57E+07	4.86E+08	5.86E-01	1.06E+07	7.06E-05	1.47E+08	5.90E-01	4.88E-01
F6	Ave	6.81E+01	2.34E+05	2.74E+06	1.86E+02	2.22E+04	3.02E+06	5.68E+05	5.41E-03	<b>7.86E-04</b>
	Std	1.57E+01	1.26E+04	5.77E+04	2.82E+00	7.68E+02	7.60E+04	2.14E+04	5.15E-03	1.48E-03
F7	Ave	4.31E-03	1.75E+03	1.99E+05	3.97E-02	2.37E+05	2.41E+05	1.68E+04	1.92E-04	<b>1.21E-04</b>
	Std	5.47E-03	1.65E+02	6.26E+03	8.98E-03	5.94E+03	3.06E+03	1.19E+03	2.03E-04	9.49E-05
F8	Ave	-3.35E+05	-8.51E+04	-8.88E+04	-1.07E+05	-6.98E+04	-4.35E+04	-1.23E+05	-4.19E+05	<b>-4.19E+05</b>
	Std	6.10E+04	1.29E+04	4.54E+03	7.03E+03	1.79E+04	6.91E+03	4.19E+03	3.32E+01	1.88E+01
F9	Ave	6.06E-14	7.60E+03	1.55E+04	1.20E+02	1.35E+04	1.70E+04	1.41E+04	<b>0.00E+00</b>	<b>0.00E+00</b>
	Std	3.32E-13	1.60E+02	2.16E+02	2.75E+01	4.64E+02	3.37E+02	3.36E+02	0.00E+00	0.00E+00
F10	Ave	4.44E-15	1.45E+01	2.03E+01	2.82E-03	1.42E+01	2.10E+01	2.10E+01	<b>8.88E-16</b>	<b>8.88E-16</b>
	Std	2.64E-15	1.75E-01	1.97E-01	2.85E-04	2.76E-01	3.42E-02	2.09E-02	0.00E+00	0.00E+00
F11	Ave	7.40E-18	2.10E+03	2.47E+04	3.02E-03	5.67E+01	2.72E+04	5.10E+03	<b>0.00E+00</b>	<b>0.00E+00</b>
	Std	2.82E-17	9.23E+01	3.86E+02	1.39E-02	9.00E+00	4.18E+02	2.63E+02	0.00E+00	0.00E+00
F12	Ave	1.12E-01	1.09E+07	3.04E+10	7.89E-01	2.86E+06	3.69E+10	2.22E+09	1.27E-06	<b>6.22E-07</b>
	Std	5.39E-02	3.52E+06	7.05E+08	3.26E-02	5.29E+05	2.21E-04	3.16E+08	2.82E-06	8.61E-07
F13	Ave	3.53E+01	1.45E+08	5.59E+10	1.03E+02	3.16E+07	6.51E+10	4.80E+09	1.14E-03	<b>2.54E-04</b>
	Std	1.27E+01	2.56E+07	1.46E+09	2.63E+00	4.38E+06	1.15E-04	4.03E+08	1.75E-03	4.42E-04

**Table 13** Results of benchmark functions (F14–F23).

Func.		WOA	SSA	MFO	GWO	PSO	DE	MVO	HHO	CHHO
F14	Ave	3.67E+00	1.23E+00	1.02E+00	2.93E+00	1.16E+00	1.86E+00	1.46E+00	1.16E+00	<b>9.98E-01</b>
	Std	4.05E+00	4.28E-01	1.28E-01	2.97E+00	4.58E-01	1.48E+00	1.26E+00	3.77E-01	1.71E-11
F15	Ave	2.37E-03	7.46E-04	1.34E-03	1.06E-03	1.59E-03	2.35E-03	3.79E-04	<b>3.30E-04</b>	4.02E-03
	Std	6.10E-03	2.51E-04	4.10E-03	1.34E-03	3.56E-03	4.91E-03	1.76E-04	2.67E-05	7.44E-03
F16	Ave	<b>-1.03E+00</b>	<b>-1.03E+00</b>	<b>-1.03E+00</b>	<b>-1.03E+00</b>	<b>-1.03E+00</b>	<b>-1.03E+00</b>	<b>-1.03E+00</b>	<b>-1.03E+00</b>	<b>-1.03E+00</b>
	Std	4.60E-09	6.78E-16	2.47E-02	9.35E-10	2.21E-14	6.78E-16	9.76E-10	9.32E-10	2.87E-07
F17	Ave	<b>3.98E-01</b>	<b>3.98E-01</b>	3.99E-01	<b>3.98E-01</b>	<b>3.98E-01</b>	<b>3.98E-01</b>	<b>3.98E-01</b>	<b>3.98E-01</b>	<b>3.98E-01</b>
	Std	1.02E-04	0.00E+00	6.16E-03	1.06E-05	5.65E-15	0.00E+00	1.27E-05	2.96E-04	2.91E-07
F18	Ave	<b>3.00E+00</b>	<b>3.00E+00</b>	3.07E+00	<b>3.00E+00</b>	<b>3.00E+00</b>	<b>3.00E+00</b>	<b>3.00E+00</b>	<b>3.00E+00</b>	<b>3.00E+00</b>
	Std	3.19E-06	4.81E-16	5.98E-01	1.47E-04	2.85E-13	1.53E-15	3.22E-07	3.66E-08	1.76E-06
F19	Ave	<b>-3.86E+00</b>	<b>-3.86E+00</b>	<b>-3.86E+00</b>	-3.85E+00	<b>-3.86E+00</b>	<b>-3.86E+00</b>	<b>-3.86E+00</b>	-3.85E+00	<b>-3.86E+00</b>
	Std	2.40E-03	2.68E-15	8.55E-03	1.80E-02	5.18E-09	1.44E-03	2.05E-03	2.75E-02	1.06E-06
F20	Ave	-3.27E+00	-3.28E+00	<b>-3.30E+00</b>	-3.20E+00	-3.22E+00	-3.23E+00	-3.12E+00	-3.13E+00	-3.25E+00
	Std	6.00E-02	5.83E-02	7.24E-02	1.78E-01	5.54E-02	5.65E-02	1.09E-01	1.07E-01	5.96E-02
F21	Ave	-9.48E+00	-7.88E+00	-9.46E+00	-8.26E+00	-7.65E+00	-6.56E+00	-5.53E+00	<b>-1.01E+01</b>	-8.47E+00
	Std	1.75E+00	2.68E+00	1.55E+00	2.50E+00	3.41E+00	3.33E+00	1.47E+00	7.40E-03	2.67E+00
F22	Ave	<b>-1.04E+01</b>	-9.52E+00	-9.92E+00	-7.60E+00	-8.20E+00	-6.60E+00	-5.25E+00	<b>-1.04E+01</b>	-8.66E+00
	Std	4.40E-04	2.00E+00	1.61E+00	3.12E+00	3.23E+00	3.65E+00	9.13E-01	8.35E-03	2.77E+00
F23	Ave	-9.99E+00	-9.82E+00	-1.00E+01	-7.55E+00	-9.10E+00	-7.87E+00	-5.03E+00	<b>-1.05E+01</b>	-8.86E+00
	Std	2.06E+00	1.85E+00	1.53E+00	3.30E+00	2.96E+00	3.39E+00	5.04E-01	1.17E-02	3.13E+00

It is very challenging for metaheuristic algorithms to solve composite mathematical functions due to their inherent challenging nature. Solving such problems requires a proper balance between its exploration and exploitation phases. As per the results in Table 14, NCHHO tends to be better than other algorithms. This demonstrates the success of using the controlling parameter in the propose algorithm has helped to balance exploration and exploitation.

In contrast, the run-time of meta-heuristic algorithms in finding the optimal solution is an important factor, which needs to be investigated to evaluate the proposed algorithm against such metric. To investigate the runtime of all algorithms on F1–F13 in seconds, Table 16 is provided. The Wilcoxon rank-sum test was implemented to statistically analyze the running time. Wilcoxon rank-sum is a nonparametric that was used in the performance analysis process as an alternative to the two-sample t-test, which is based solely on the order in which the observations from the two samples fall. This test is therefore used to evaluate the results of the proposed method with other techniques in pairs. on inspecting Table 16, it is evident that the NCHHO has an acceptable running time. However, the use of chaotic values had little effect on the time that was enquired in finding the best global solution relative to HHO. On analyzing the captured results, it is observed that the NCHHO algorithm could

achieve a reasonable and competitive running time as compared to other optimizers in dealing with unimodal and multimodal functions, even with high-dimensional solution search space. This performance is due to the unique structure of metaheuristics in dealing with the optimization problems.

**Table 14** Results for composition functions (CEC2014)[70].

Func.		GWO	PSO	DE	MVO	SSA	MFO	WOA	HHO	NCHHO
F24	Ave	2.64E+03	2.61E+03	2.62E+03	2.62E+03	2.63E+03	2.67E+03	2.69E+03	2.50E+03	<b>2.50E+03</b>
	Std	1.10E+01	8.66E-05	1.72E-03	5.46E+00	9.15E+00	3.93E+01	2.23E+01	1.20E-06	0.00E+00
F25	Ave	2.60E+03	2.62E+03	2.63E+03	2.64E+03	2.64E+03	2.68E+03	2.61E+03	2.60E+03	<b>2.60E+03</b>
	Std	1.12E-02	7.36E+00	1.70E+00	7.08E+00	6.59E+00	2.98E+01	8.25E+00	1.03E-03	<b>5.24E-06</b>
F26	Ave	2.71E+03	2.72E+03	2.72E+03	2.71E+03	2.72E+03	2.72E+03	2.72E+03	2.70E+03	2.70E+03
	Std	6.00E+00	4.12E+00	3.00E+00	2.26E+00	4.86E+00	1.01E+01	2.03E+01	8.03E-05	0.00E+00
F27	Ave	2.74E+03	2.77E+03	<b>2.70E+03</b>	2.74E+03	2.70E+03	2.71E+03	2.73E+03	2.77E+03	2.77E+03
	Std	4.96E+01	4.48E+01	7.86E-02	5.92E+01	1.38E-01	4.95E+01	6.52E+01	4.77E+01	4.38E+01
F28	Ave	3.43E+03	3.59E+03	3.49E+03	3.35E+03	3.60E+03	3.63E+03	3.85E+03	2.90E+03	<b>2.90E+03</b>
	Std	1.32E+02	2.72E+02	1.27E+02	1.68E+02	1.73E+02	1.84E+02	3.81E+02	9.14E-05	0.00E+00
F29	Ave	4.04E+03	6.72E+03	3.73E+03	4.10E+03	4.14E+03	3.92E+03	5.43E+03	<b>3.00E+03</b>	<b>3.00E+03</b>
	Std	3.24E+02	6.03E+02	3.65E+01	3.45E+02	3.60E+02	1.22E+02	8.53E+02	0.00E+00	0.00E+00

**Table 15** Results of Wilcoxon rank-sum test over all runs.

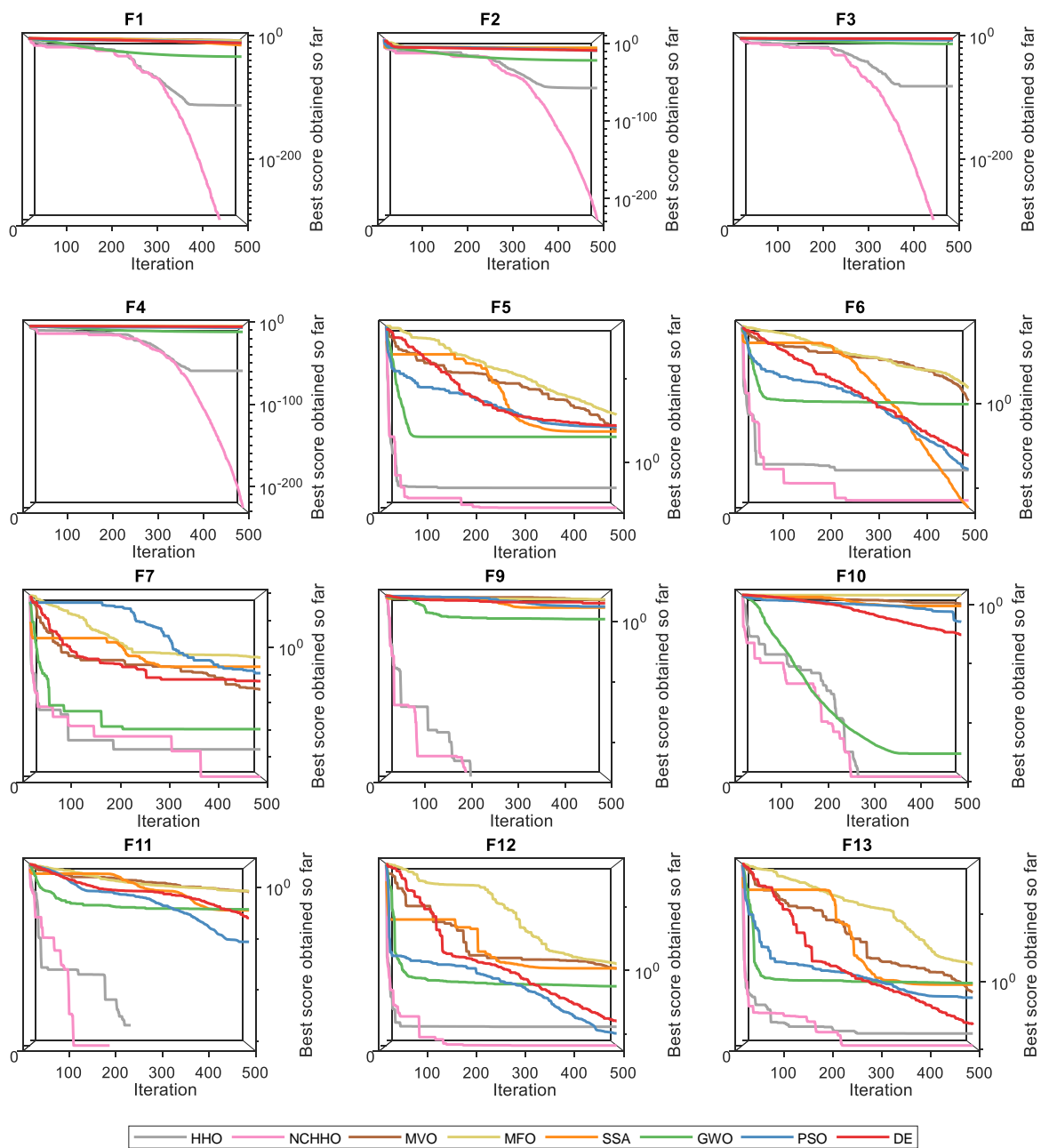
Func.	WOA	SSA	MFO	GWO	PSO	DE	MVO	HHO
F1	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
F2	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
F3	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
F4	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
F5	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	4.83E-01
F6	3.02E-11	5.00E-09	3.02E-11	7.60E-07	3.52E-07	3.69E-11	3.02E-11	4.64E-05
F7	1.07E-09	3.02E-11	3.02E-11	4.08E-11	3.02E-11	3.02E-11	3.02E-11	2.84E-01
F8	4.50E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	6.52E-01
F9	3.34E-01	1.21E-12	1.21E-12	4.55E-08	1.21E-12	1.21E-12	1.21E-12	NaN
F10	3.86E-09	1.21E-12	1.21E-12	4.49E-13	1.21E-12	1.21E-12	1.21E-12	NaN
F11	NaN	1.21E-12	1.21E-12	1.61E-01	1.21E-12	1.21E-12	1.21E-12	NaN
F12	3.02E-11	3.02E-11	3.02E-11	6.70E-11	8.56E-04	3.34E-11	3.02E-11	1.86E-03
F13	3.02E-11	3.02E-11	3.02E-11	5.00E-09	5.46E-06	3.47E-10	3.02E-11	3.27E-02

**Table 16** Comparison of average runtime of all algorithms in seconds over 30 runs.

Func.		WOA	SSA	MFO	GWO	PSO	DE	MVO	HHO	CHHO
F1	Ave	5.79E+00	1.33E+02	1.28E+01	1.85E+01	6.84E+00	1.64E+02	1.74E+01	3.99E+01	4.54E+01
	Std	2.35E-01	9.42E+00	1.14E+00	1.97E+00	3.10E-01	3.67E+00	1.43E+00	2.51E+00	3.27E+00
F2	Ave	7.32E+00	1.26E+02	1.26E+01	1.74E+01	8.22E+00	1.85E+02	1.60E+01	3.99E+01	4.22E+01
	Std	5.75E-01	5.79E+00	9.51E-01	1.04E+00	7.13E-01	2.86E+00	9.25E-01	3.27E+00	3.37E+00
F3	Ave	2.85E+01	1.36E+02	3.27E+01	3.84E+01	2.85E+01	2.03E+02	3.30E+01	9.32E+01	8.63E+01
	Std	2.24E+00	7.18E-01	2.46E+00	3.00E+00	1.90E+00	7.21E+00	5.72E-01	7.03E+00	8.44E+00
F4	Ave	6.29E+00	1.16E+02	1.30E+01	1.85E+01	7.12E+00	1.62E+02	1.52E+01	4.91E+01	4.01E+01
	Std	5.71E-01	2.08E+00	1.31E+00	1.14E+00	6.94E-01	7.82E-01	2.45E-01	1.13E+00	1.02E+00
F5	Ave	8.84E+00	1.14E+02	1.54E+01	2.03E+01	8.98E+00	1.68E+02	1.76E+01	6.50E+01	6.75E+01
	Std	2.87E-01	1.84E+00	2.22E+00	3.32E+00	2.14E-01	1.09E+00	2.94E-01	8.05E+00	7.28E+00
F6	Ave	5.90E+00	1.16E+02	1.25E+01	1.77E+01	6.74E+00	1.64E+02	1.55E+01	5.35E+01	5.73E+01
	Std	2.30E-01	6.83E-01	9.04E-01	1.19E+00	3.09E-01	1.13E+00	3.83E-01	2.32E+00	1.76E+00
F7	Ave	1.61E+01	1.28E+02	2.35E+01	2.83E+01	1.66E+01	1.73E+02	2.50E+01	6.85E+01	6.95E+01
	Std	4.54E-01	8.05E-01	1.31E+00	1.41E+00	2.35E-01	6.87E-01	2.94E-01	2.16E+00	2.14E+00
F8	Ave	8.91E+00	1.19E+02	1.59E+01	2.14E+01	1.06E+01	1.68E+02	1.51E+01	6.32E+01	6.58E+01
	Std	5.09E-01	9.40E-01	1.18E+00	1.47E+00	5.42E-01	7.60E-01	2.61E-01	1.59E+00	1.76E+00
F9	Ave	6.91E+00	1.19E+02	1.35E+01	1.71E+01	8.78E+00	1.64E+02	1.73E+01	5.93E+01	6.06E+01
	Std	6.50E-01	6.41E-01	3.72E-01	3.06E-01	6.89E-01	7.15E-01	2.95E-01	2.42E+00	3.06E+00
F10	Ave	7.29E+00	1.19E+02	1.41E+01	1.75E+01	8.84E+00	1.68E+02	1.77E+01	6.02E+01	6.07E+01
	Std	5.66E-01	1.23E+00	2.65E-01	4.17E-01	5.77E-01	4.36E-01	2.99E-01	1.46E+00	1.59E+00
F11	Ave	9.28E+00	1.25E+02	1.73E+01	2.12E+01	1.06E+01	1.71E+02	2.03E+01	6.40E+01	6.53E+01
	Std	5.78E-01	5.88E+00	6.17E-01	5.05E-01	7.18E-01	5.98E-01	1.03E+00	1.61E+00	1.96E+00
F12	Ave	3.79E+01	1.45E+02	4.18E+01	4.54E+01	3.90E+01	2.18E+02	4.28E+01	1.24E+02	1.31E+02
	Std	3.02E+00	1.47E+00	5.98E-01	4.04E-01	2.25E+00	1.26E+01	2.95E-01	4.03E+00	4.17E+00
F13	Ave	3.80E+01	1.45E+02	4.18E+01	4.57E+01	3.95E+01	2.25E+02	4.45E+01	1.24E+02	1.30E+02
	Std	1.45E+00	7.10E+00	3.62E-01	4.33E-01	1.77E+00	8.73E+00	1.88E+00	1.22E+01	4.45E+00

In order to analyze the convergence behavior of the proposed NCHHO algorithm, the best score metric obtained so far was defined and measured as the obtained optimal value. This score was

calculated and captured for each particular iteration of all utilized test functions. This value was calculated using each implemented and benchmarked method and cross-checked along with the proposed algorithm's average performance that captured the course of 30 simulation runs with 500 iterations for each. In the convergence debate, as shown in Fig. 7, the proposed method has, in most cases, an acceptable convergence rate compared to other methods. The reported results have demonstrated and proved the efficiency of using chaos values, which has dramatically improved the performance and the convergence rate of the optimization algorithm that is specifically represented by the proposed method. In some of the functions, however, NCHHO achieved the optimum values that were almost followed by the HHO standard, which can be observed in sub-figures of results of F7, F9 and F10. The NCHHO algorithm not only outperformed the other algorithms in the convergence curve, but also has achieved an optimum value in early iterations, and showed a very remarkable convergence speed compared to other algorithms, as can be seen from results visualized in Fig. 7 of F5, F6, F11 and F12. While it is evident with other benchmarked algorithms, have been stacked with the local optimal solutions from the early stage of the experiments, which can be observed by results of F1-F7 as well as some of the other test functions.



**Fig.7.** Convergence analysis of the NCHHO algorithm compared with other techniques.

In the above paragraphs, we discussed the performance of the proposed method in details. In the following paragraphs, other comparative algorithms are discussed and analyzed, including HHO, MVO, MFO, SSA, GWO, PSO, and DE. Considering the convergence curves in Fig. 7 and tables, the HHO algorithm tends to be the second best on the majority of test functions. The convergence behavior is quite similar to that of NCHHO, but it tends to flatten as the optimization process progresses. This shows that such a pattern has been improved using the proposed mechanisms in NCHHO.

The PSO algorithm is usually the third best algorithm on the majority of case studies. The PSO algorithm uses GBEST and PBESTs to update the position of particles, so there are several “guides” for particles, which resulted in showing competitive performance by this algorithm. PSO is one of the best algorithms in the literature and its superiority is evident in the results. However, our results evidently show that the proposed NCHHO provides better performance compared to this algorithm. The performance of the DE algorithm is quite similar to PSO as can be seen in Fig. 7 and Tables 10-16. Despite similarities, DE uses several mutation operators that leads to sudden changes in the solutions. This is a good mechanism for exploration but negatively impacts exploitation. The results show that DE’s performance is quite consistent and rarely show accelerated or decelerated convergence. This is because the lack of adaptive mechanism in this algorithm.

GWO’s average performance comes after PSO. This algorithm shows quick convergence and low improvement as the iteration counter increases (See Fig. 7). The reason for this is because GWO uses only three best solutions obtained so far (alpha, beta, and delta) to update the position of other wolves. This leads to lower exploration but higher exploitation than PSO. The GWO algorithm uses an adaptive mechanism so it shows better results than DE. The SSA algorithm provides very competitive results compared to GWO. This algorithm also uses one leader to update the position of all solutions during the optimization process, which leads to less exploratory behavior compared to GWO. This is why this algorithm shows low convergence and less accurate results than the GWO algorithm. Overall, it is evident that the GWO algorithm is also outperformed by the proposed NCHHO algorithm.

MVO and MFO algorithms ranked the lowest on most case studies. The main reason for this is potentially due to the small number of adaptive parameters in both algorithms to effectively tune exploration and exploitation. This can be supported by looking at the results in the above tables and Fig. 7. It is quite obvious that the convergence behavior of both algorithms is monotonous, which is not desired when an algorithm requires to accelerate or decelerate the pace of changes in solutions to avoid locally optimal solutions and achieve a good estimation of the global optimum.

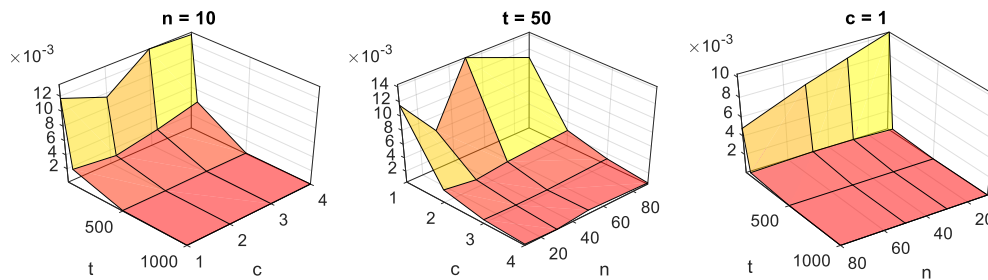
### 5.5. Sensitivity Analysis

As any other new algorithm, NCHHO has several controlling parameters. The analysis of such key, internal parameters such as the number of search agents, the maximum number of iterations, and the control parameter specified in Eq. (15) is essential to ensure maximum performance. In this subsection, a sensitivity analysis is conducted to investigate the impact of the main controlling parameters of the proposed method on its performance. The following values are used:

- Number of search agents (N): 10, 30, 50, 80
- Maximum iteration number (T): 50, 100, 500, or 1000
- Nonlinear control parameter  $c$ : Nonlinearly decreases from 1 to 0, 2 to 0, 3 to 0, or 4 to 0

In order to demonstrate the effect of these parameters on the performance of NCHHO, three independent experiments were carried out by simultaneously changing the values of the three parameters listed above. To be able to visualize the performance surface in each experiment, one of

the parameters is considered fixed while the other two keep changing. NCHHO is run 30 times on four test functions (F1, F2, F12, and F13), and the average of the best solution in the last iteration are used to visualize the average performance surface in Fig. 8.

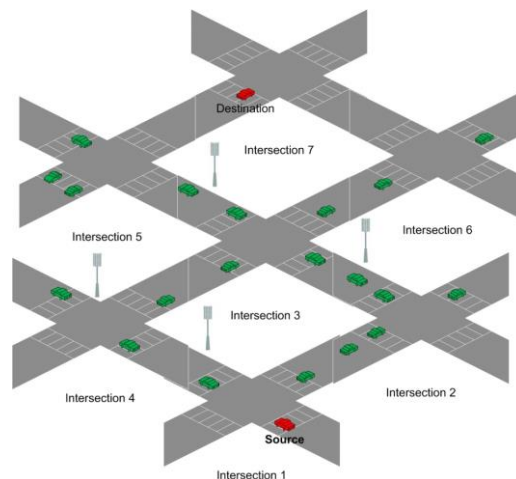


**Fig. 8.** Sensitivity analysis experiment. Each surface shows the average performance surface of NCHHO.

Fig. 8 shows that the increase in the average performance of NCHHO (low objective values due to minimization) is proportional to the number of iterations and solutions. However, there is no significant performance improvement when  $t > 500$  and  $n > 50$ . A similar observation can be made for the parameter  $c$ , in which the performance increases proportional to the value of  $c$ .

#### 5.6. Solving Internet of Vehicles Routing Problem

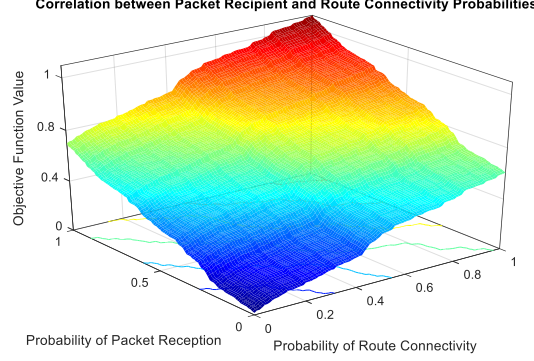
In order to evaluate the performance of NCHHO, an IoV related optimization problem was formulated and implemented. Fig. 9 shows the IoV scenario with two highlighted vehicles, where one acts as the source of the network traffic, and the other is the destination where the traffic must be delivered. From the point view of the source vehicle, there are multiple options available to form the optimal route to the destination vehicle, which could involve multiple intersections. Each route offers different Quality of Service (QoS) value, which is highly correlated to several performance measures. The objective of this problem is to maximize the probability of connectivity and link QoS of the available routes from source to destination as illustrated in Fig. 9.



**Fig. 9.** Internet of Vehicles (IoV) Scenario: Finding the optimal network route problem.

The maximization method is subject to the probabilities of route connectivity ( $P_c$ ) and packet reception ( $P_r$ ) of each route, taking into the consideration the constraint of route delay ( $D$ ). Fig. 10 demonstrates the correlation between the probabilities of route connectivity and packet reception as well as their impact on the obtained objective function values when a liner increment is applied on each metric; the higher the  $P_c$  and  $P_r$ , the maximum the objective function value is. We can observe that when a route segment is offering high probabilities of  $P_c$  and  $P_r$ , the objective function value will be close to 1 (Maximum value) and vice versa. In real life scenario, each communicating vehicle

offers different probability of connectivity and packet reception. Taking into the consideration of the different delay values offered by each available route from source to destination, (which we noted as route delay  $D$ ), selecting the best combination of these parameters among all available delay values offered by vehicles in a road segment as an optimal route with maximum objective value, it tending to be a non- polynomial (NP) problem.



**Fig. 10.** Objective function's behavior of IoV optimization problem with respect to  $P_c$  and  $P_r$ .

Therefore, NCHHO, along with the other benchmark optimization algorithms, is used in order to solve the optimization problem of finding the more reliable and connected route in urban software defined network (SDN) based vehicular communication to enable IoV application.

The city road networks in vehicular scenario was denoted as graph model  $G(i,s)$  where  $i$  is an intersection and  $e$  is the road segment between two intersections. Hence, each optimal route  $\alpha$  consists of a set of intersections ( $i_1; i_2; i_3; i_4; i_5; i_6; \dots; i_m$ ) and a set of streets ( $s_1; s_2; s_3; s_4; s_5; s_6; \dots; s_n$ ), where  $n=m-1$ . According to the aforementioned assumptions, the objective function of the optimization problem can be written as:

$$Max(\alpha) = F(\alpha) = \gamma_1 \times P_c(\alpha) + \gamma_2 \times P_r(\alpha) + \gamma_3 \times D(\alpha) \times \frac{1}{1 + T_v} \quad (22)$$

$$Where P_c(\alpha) = \prod_{i=1}^n P_c(e_i)$$

$$Where P_r(\alpha) = \prod_{i=1}^n P_r(e_i)$$

$$Where D(\alpha) = \frac{\sum_{i=1}^n D_{th}(e_i) - \sum_{i=1}^n D(e_i)}{\sum_{i=1}^n D_{th}(e_i)}$$

Where  $D_{th}(e_i)$ , the Delay threshold value

$$D_{th}(e_i) = \frac{\sum_{i=1}^n D(e_i)}{2}$$

Subject to  $D(e_i) \leq D_{th}(e_i)$

$T_v$  is the time variation of entire graph model of a network map  $G(i,s)$

if size of  $G(i,s) = 1000$ , then

$$T_v = \sum_{i=1}^n D(e_i) / 1000$$

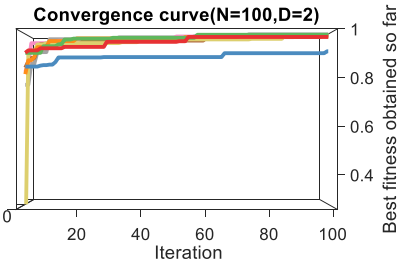
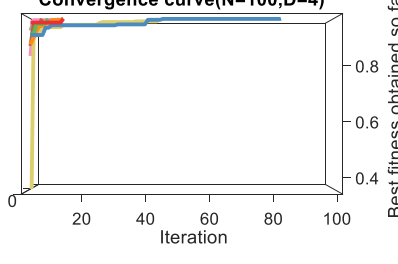
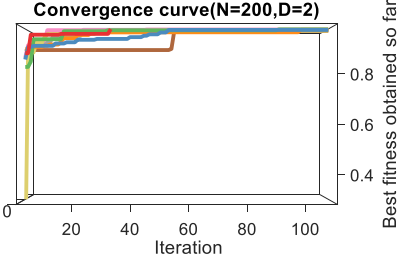
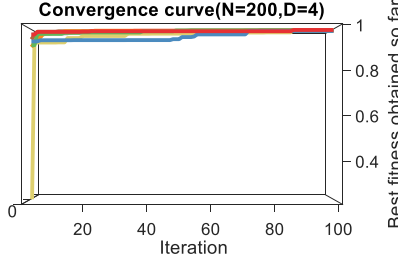
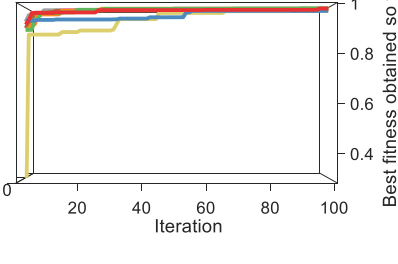
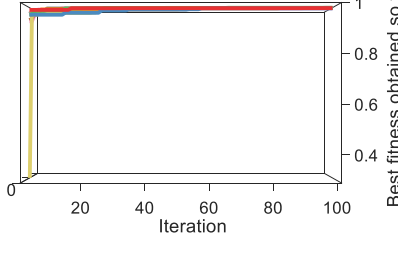
The IoV problem was addressed by applying a set of heuristic algorithms. The proposed algorithm has been applied along with the other benchmark algorithms to find the optimal route from source to destination vehicles. It is important to mention that for each created IoV problem, initial random values were created for the selected dimensions ( $P_c$ ,  $P_r$ ,  $D$ , and  $T_v$ ). For each route solution, these

values will be captured, and each algorithm will search the optimal route that could offer the maximum value of the objective function  $F(\alpha)$ . The probability of connectivity for each available route of a road segment to the destination vehicle  $P_c(e_i)$  is calculated and integrated with probability of packet reception  $P_r(e_i)$  and route delay, as presented by Eq. 22. In order to have a more reliable decision, the delay metric was captured and compared with a threshold of each defined road segment  $D_{th}(e_i)$ , where this threshold is the average delay experienced by a particular route set  $e_i$ . A time variance  $T_v$  was introduced also to increase the reliability of the chosen optimal route.

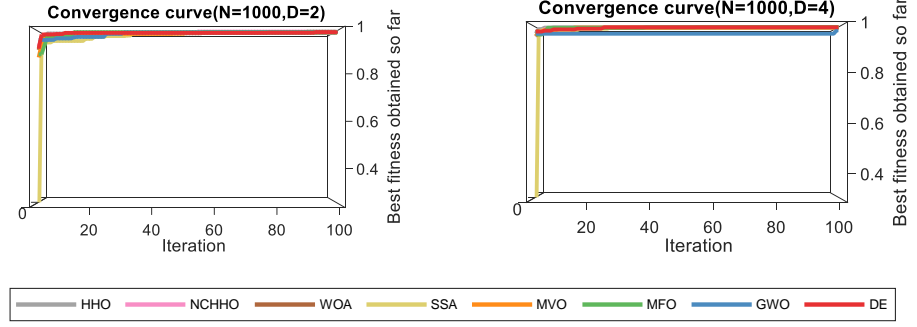
Another experiment was set and the NCHHO algorithm was implemented along with the other benchmark algorithms WOA, PSO, DE, SSA, MVO, GWO, MFO, and HHO, to solve the formulated IoV optimization problem. Table 15 has listed the captured convergence rates of the best value achieved out of the objective function (Eq. 22) over the course of 500 iterations. It is essential to highlight that the obtained results were an average of 50 simulation runs for each optimization algorithm. We have investigated the ability of each algorithm in optimizing different dimensions of the IoV problem.

The second column in Table 17 lists the graphs of the obtained convergence rates using only two dimensions of the problem, which are  $P_c$  and  $P_r$ . This experiment was implemented with four different sets of graph models ( $100 \times 100$ ,  $200 \times 200$ ,  $500 \times 500$  and  $1000 \times 1000$ ). We have observed that the NCHHO algorithm could manage a good convergence compared with the other implemented algorithms, though in terms of STD over an average of 50 runs; DE and HHO have maintained a slightly higher STD value when the graph size was set to  $N=100$  and  $200$ ; while MFO could achieve a slightly better STD value with  $N=500$  and  $1000$  (as listed by Table 17 accordingly).

**Table 17** Convergence analysis of the NCHHO algorithm compared with other techniques obtained over 50 runs by 2 and 4 dimensions IoV problems with four different  $N$  sizes.

N/Dim Values		
Graph model Size (N)	IoV with 2 Problem Dimensions ( $P_c$ and $P_r$ )	IoV with 4 Problem Dimensions ( $P_c, P_r, D$ and $T_v$ )
$100 \times 100$	 <p style="text-align: center;">Convergence curve(N=100,D=2)</p>	 <p style="text-align: center;">Convergence curve(N=100,D=4)</p>
$200 \times 200$	 <p style="text-align: center;">Convergence curve(N=200,D=2)</p>	 <p style="text-align: center;">Convergence curve(N=200,D=4)</p>
$500 \times 500$	 <p style="text-align: center;">Convergence curve(N=500,D=2)</p>	 <p style="text-align: center;">Convergence curve(N=500,D=4)</p>

1000×1000



This also reflects the fact that having only  $P_c$  and  $P_r$  parameters could not reflect the cost of the objective value to achieve its optimal maximum value perfectly. Hence, we have included in our second experiment four dimensions of the IoV problem ( $P_c$ ,  $P_r$ ,  $D$ , and  $T_v$ ). This slightly better STD measures of the performance of DE, HHO, and MFO were obtained when the IoV problem was with only two dimensions ( $P_c$  and  $P_r$ ), whereas the proposed NCHHO algorithm performed the best with high STD, when the problem dimension was set to four ( $P_c$ ,  $P_r$ ,  $D$ , and  $T_v$ ). From this observation we could interpret that NCHHO performs better with an optimization problem that has high dimensions in comparison with other benchmarked algorithms, as can also be seen from the given STD values in Table 17. The reason behind this could be the nonlinearity and chaotic behaviors that the algorithm has gained from the proposed improvement to the standard HHO algorithm.

**Table 17** Results obtained over 50 runs by different IoV problem’s dimensions and variables of graph sizes.

Dimension = 2 , N = 100×100				Dimension = 2 , N = 200×200			
Algorithm	Best	AVG	STD	Algorithm	Best	AVG	STD
HHO	1	0.9963	2.08E-03	HHO	1	0.9970	1.17E-03
NCHHO	1	0.9971	1.94E-03	<b>NCHHO</b>	<b>1</b>	<b>0.9981</b>	<b>1.10E-03</b>
WOA	1	0.9969	2.33E-03	WOA	1	0.9976	1.19E-03
SSA	0.9994	0.9951	2.68E-03	SSA	1	0.9956	2.27E-03
MVO	1	0.9966	2.41E-03	MVO	1	0.9974	2.21E-03
<b>MFO</b>	<b>1</b>	<b>0.9971</b>	<b>1.69E-03</b>	MFO	1	0.9980	1.11E-03
GWO	1	0.9947	3.34E-03	GWO	0.9996	0.9971	1.12E-03
DE	0.9995	0.9968	1.68E-03	DE	0.9996	0.9972	1.59E-03
Dimension = 2 , N = 500×500				Dimension = 2 , N = 1000×1000			
Algorithm	Best	AVG	STD	Algorithm	Best	AVG	STD
HHO	1	0.9987	9.36E-04	HHO	1	0.9994	5.84E-04
NCHHO	1	0.9992	8.13E-04	NCHHO	1	0.9995	6.33E-04
WOA	1	0.9989	9.23E-04	WOA	1	0.9993	6.45E-03
SSA	1	0.9983	1.39E-03	SSA	1	0.9991	8.63E-04
MVO	1	0.9983	1.19E-03	MVO	1	0.9991	9.26E-04
<b>MFO</b>	<b>1</b>	<b>0.9992</b>	<b>7.55E-04</b>	MFO	1	0.9995	5.33E-04
GWO	1	0.9983	1.29E-03	GWO	1	0.9989	8.66E-04
DE	1	0.9989	9.14E-04	<b>DE</b>	1	<b>0.9997</b>	<b>5.46E-04</b>
Dimension = 4 , N = 100×100				Dimension = 4 , N = 200×200			
Algorithm	Best	AVG	STD	Algorithm	Best	AVG	STD
HHO	1	0.9991	7.50E-04	HHO	1	0.9994	9.18E-04
<b>NCHHO</b>	<b>1</b>	<b>0.9992</b>	<b>7.41E-04</b>	<b>NCHHO</b>	<b>1</b>	<b>0.9996</b>	<b>5.39E-04</b>
WOA	1	0.9985	1.19E-03	WOA	1	0.9993	6.33E-04
SSA	1	0.9982	1.54E-03	SSA	1	0.9989	1.14E-03
MVO	1	0.9984	1.19E-03	MVO	1	0.9989	8.95E-04
MFO	1	0.9991	8.17E-04	MFO	1	0.9994	6.36E-04
GWO	1	0.9980	1.49E-03	GWO	1	0.9984	1.09E-03
DE	1	0.9991	8.13E-04	DE	1	0.9994	6.44E-04
Dimension = 4 , N = 500×500				Dimension = 4 , N = 1000×1000			
Algorithm	Best	AVG	STD	Algorithm	Best	AVG	STD
HHO	1	0.9998	5.76E-04	HHO	1	0.9999	4.42E-04
<b>NCHHO</b>	<b>1</b>	<b>1</b>	<b>4.33E-04</b>	<b>NCHHO</b>	<b>1</b>	<b>1</b>	<b>2.31E-04</b>
WOA	1	0.9999	4.63E-04	WOA	1	1	2.82E-04

SSA	1	0.9996	5.41E-04	SSA	1	0.9999	4.32E-04
MVO	1	0.9996	4.63E-04	MVO	1	0.9998	5.28E-04
MFO	1	0.9999	4.53E-04	MFO	1	1	2.57E-04
GWO	1	0.9994	6.13E-04	GWO	1	0.9999	4.01E-04
DE	1	0.9999	3.89E-04	DE	1	1	2.38E-04

It was clear that the proposed NCHHO algorithm could obtain the best among the other benchmarked algorithms, in terms of convergence ability as listed in the AVG column in Table 17. On the other hand, for reliability analysis, as STD values were listed in Table 17; the proposed NCHHO algorithm has managed better STD values as compared to other algorithms. This reflects the ability of the NCHHO algorithm to handle optimization problems with relatively high dimensions and graph size of IoV routing problem.

As per these results and discussions, we would like to discuss the contributions of this work once again. The use of chaotic map in HHO instead of random components to improve its exploration was demonstrated by the comparative experiments on the rest functions. The results showed that the chaotic maps improve the exploratory behavior of HHO without increasing its computational cost. While the use of chaotic values is successful in improving the algorithm's exploration process, it may create a disruption between the exploration phase and the algorithm's exploitation. The results showed that the integration of the non-linear mechanism to HHO is a remedy for this. Finally, the last contribution was the use of the proposed NCHHO algorithm in solving one of the challenging optimization problems in the area of wireless communication for IoV as a seminal attempt to tackle this challenging problem using metaheuristics.

### 5.7 Results analysis and discussion

To sum up, the results showed that the original version of the HHO algorithm shows poor performance on challenging test problems and real-world case studies with a wide range of difficulties. This observation can be made in other metaheuristics too, which is due to the fact that such techniques are designed general-purpose enough to solve different classes of optimization problems. This makes them applicable to many problems, however, solving a particular set of problems requires algorithm modification and tuning. Looking at the results in the preceding sections, it is evident that the standard HHO and the compared bunch of recent optimization methods are not suitable algorithms for large-scale problems with such dynamic number of variables. The case study considered in this work is in the area of IoV Application, and therefore, algorithm modifications were essential.

We purposely considered problems with a larger number of variables to stress test the HHO algorithm, and it was observed that the performance of this algorithm substantially decreased in proportion to the scale of the problem. This is due to the exponential growth of the search space in large-scale optimization problems, that require intensive exploration in a metaheuristic. There are many methods in the literature to improve exploratory behavior of metaheuristics, however, majority of them increase the computation cost of the algorithm too. This makes them less practical for computationally expensive optimization problems, including the one investigated in this work. Therefore, we used chaotic map as one of the most computationally cheap techniques to improve exploration.

On the other hand, improving the exploration of an algorithm tends to disrupt its accuracy. This is because when an algorithm explores a search space, it discovers new regions without using any local search mechanism. Our initial investigations and results showed that using only chaotic maps in HHO would lead to this issue. Therefore, we proposed the use of a non-linear parameter too, which assists HHO with improving its accuracy.

The results on test functions demonstrated that chaotic map and non-linear mechanism can play a significant role in improving the performance of HHO. In the majority of test functions, the proposed NCHHO algorithm outperformed HHO. The discrepancy of the results was higher and more evident on large-scale test functions, which was the main intention of this paper, for improving HHO. The superiority of NCHHO is due to its high exploratory mechanism using chaotic maps, and the smooth

transition to exploitation using the non-linear mechanism. The results on the IoV demonstrated that the proposed NCHHO algorithm is capable of solving this problem with high accuracy, and outperform HHO as well as other recent optimization algorithms significantly.

It should be noted that most of the HHO's operators have been maintained in NCHHO, so the proposed method is still general-purpose enough to solve a wide range of problems. What makes NCHHO different is its ability to better solve large-scale problems. Therefore, we offer this algorithm to researchers and practitioner to be used in such problems when HHO shows poor performance.

## 6. Conclusion and Future Work

This work proposed a chaotic and non-linear version of the HHO algorithm. The article started with a comprehensive review of chaotic metaheuristics and related works in the literature. The gap was identified as the inefficiency of HHO when applied to large-scale problems, despite being a general-purpose, black-box optimizer. Several initial experiments were conducted to find the reason of degraded accuracy in HHO when solving large-scale problems. It was observed that the exploration of HHO is its main weaknesses for such problems.

After identifying the issue, several chaotic maps were integrated into the HHO algorithm in an attempt to find the best one for maximizing the exploration. The best chaotic map was found, however, it was observed that the accuracy of the chaotic HHO was negatively impacted. To combat this, a non-linear parameter was proposed and integrated into this algorithm. The algorithm was called Non-linear Chaotic Harris Hawk Optimizer (NCHHO) and compared to a wide range of techniques in the literature on a set of test functions. The advantage of this method is the use of a control parameter to create a proper balance between the exploration and exploitation phases. In addition, it improves the exploration phase of the HHO algorithm by using chaotic values instead of random values.

NCHHO was tested on a wide range of unimodal, multimodal, and composite test functions with a diverse number of parameters. The results demonstrated the superiority of NCHHO as compared to HHO and other comparative algorithms on most test cases. The use of the controlling parameter resulted in having a good balance between exploration and exploitation. The paper also considered parameter-setting analysis of NCHHO to observe the impact of the main controlling parameters of its performance. In the end, the proposed method was also employed, and its performance was measured to find optimal solutions for a challenging problem in the area of IoV routing problem. NCHHO showed promising results in finding the optimal route for Internet of Vehicles application. The proposed algorithm provided substantially better results as compared to the state-of-the-art algorithm, which is a strong indication of its merits.

With the proposed mechanisms, the work contributed to the literature by using chaotic maps in HHO instead of random components, to improve its exploration. Then, we leveraged chaotic maps to improve the exploratory behavior of HHO without increasing its computational cost, which was identified as one of the shortcomings of the existing chaotic metaheuristics. While the use of chaotic values was successful in improving the algorithm's exploration process, it also led to a disruption between the exploration phase and the algorithm's exploitation, which motivated our attempt to propose a nonlinear parameter for the first time in the literature for the HHO algorithm. The last contribution was the use of NCHHO to solve one of the challenging optimization problems in the area of wireless communication for IoV as a seminal attempt to tackle this challenging problem using metaheuristics. This contribution allowed obtaining a cost-effective robust solution for the field of autonomous vehicles to improve data dissemination over a reliable optimal route.

It is worth noting that there are also some limitations in the proposed method. First, the NCHHO algorithm requires special mechanism to cope with constraints in heavily constrained problems. Second, there should be modification in this algorithm at the algorithmic level to solve mixed-integer optimization problems. Third, to handle multiple objectives, we need to either aggregate them into one objective and solve it using NCHHO or develop a multi-objective version of NCHHO. Finally, a hyper-heuristic mechanism might be required to find optimal controlling parameters for the proposed algorithm.

For future work, it is recommended to test the impact of chaotic map on the multi-objective optimization algorithms. In multi-objective optimization, two primary objectives of a posteriori method are to provide accurate estimation of Pareto optimal solutions (convergence) with high distribution (coverage). Chaotic maps have the potential to properly balance these two conflicting goals in an algorithm, which is recommended as a future direction. Optimization problems with constraints are quite common, so another interesting future direction will be to explore the impact of different constraint handling techniques (e.g., penalty function, barrier function, etc.) on the performance of NCHHO. The chaotic maps used in this work were not adaptive, so it is recommended to adaptively change their range to provide targeted exploratory and exploitative behaviors for NCHHO as well.

## References

- [1] H. Faris *et al.*, An intelligent system for spam detection and identification of the most relevant features based on evolutionary Random Weight Networks, *Inf. Fusion*, 48 (2019), pp. 67–83..
- [2] Q. Tu, X. Chen, and X. Liu, Multi-strategy ensemble grey wolf optimizer and its application to feature selection, *Appl. Soft Comput.*, 76 (2019), pp. 16–30.
- [3] A. A. Ewees, M. Abd Elaziz, and E. H. Houssein, Improved grasshopper optimization algorithm using opposition-based learning, *Expert Syst. Appl.*, 112 (2018), pp. 156–172.
- [4] A. A. Heidari and P. Pahlavani, An efficient modified grey wolf optimizer with Lévy flight for optimization tasks, *Appl. Soft Comput.*, 60 (2017), pp. 115–134.
- [5] M. A. El Aziz, A. A. Ewees, and A. E. Hassanien, Multi-objective whale optimization algorithm for content-based image retrieval., *Multimed Tools Appl*, 77 (2018), pp. 26135–26172.
- [6] A. Asghar, S. Mirjalili, H. Faris, and I. Aljarah, Harris hawks optimization : Algorithm and applications, *Futur. Gener. Comput. Syst.*, 97 (2019), pp. 849–872.
- [7] J. Dreo, A. Petrowski, P. Siarry, and E. Taillard, *Metaheuristics for Hard Optimization: Methods and Case Studies.*, Ger. BerlinSpringer, 2008.
- [8] E. Talbi, *Metaheuristics: from Design to Implementation.*, John Wiley Sons, Hoboken, 2009.
- [9] I. Rechenberg, Evolution Strategy: Nature’s Way of Optimization. In: Bergmann H.W. (eds) *Optimization: Methods and Applications, Possibilities and Limitations.*, *Lect. Notes Eng.*, 47, (1989).
- [10] D. E. Goldberg and J. H. Holland, Genetic Algorithms and Machine Learning., *Mach. Learn*, 3 (1988), no. 2, pp. 95–99.
- [11] R. Storn and K. Price, Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces., *J. Glob. Optim.*, 11 (1997), pp. 341–359.
- [12] D. Simon, Biogeography-Based Optimization., *IEEE Trans. Evol. Comput.*, 12 (2008), no. 6, pp. 702–713.
- [13] A. Coloni, M. Dorigo, and V. Maniezzo, Distributed optimization by ant colonies. In: *Proceedings of the first European conference on artificial life*, pp. 134–142, 1991.
- [14] R. Eberhart and J. Kennedy, A new optimizer using particle swarm theory, *Proc. Sixth Int. Symp. Micro Mach. Hum. Sci.*, pp. 39–43, 1995.
- [15] S. Mirjalili, A. H. Gandomi, S. Zahra, and S. Saremi, Salp Swarm Algorithm : A bio-inspired optimizer for engineering design problems, *Adv. Eng. Softw.*, 114 (2017), pp. 1–29.
- [16] Yang, Xin-She, and S. Deb, Cuckoo Search via Lévy flights., *World Congr. Nat. Biol. Inspired Comput.*, pp. 210–214, 2009.
- [17] Wen-Tsao Pan, A new Fruit Fly Optimization Algorithm: Taking the financial distress model as an example, *Knowledge-Based Syst.*, 26 (2012), pp. 69–74.
- [18] N. F. A. Kaveh, A new optimization method: Dolphin echolocation,”*Adv. Eng. Softw.*, 59 (2013), pp. 53–70.
- [19] S. Mirjalili, S. M. Mirjalili, and A. Lewis, Grey Wolf Optimizer, *Adv. Eng. Softw.*, 69 (2014), pp. 46–61.
- [20] X.-S. Yang, A New Metaheuristic Bat-Inspired Algorithm. In: González J.R., Pelta D.A., Cruz C., Terrazas G., Krasnogor N. (eds) *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*. Studies in Computational Intelligence, 284 (2010).
- [21] S. Mirjalili and A. Lewis, The Whale Optimization Algorithm, *Adv. Eng. Softw.*, 95 (2016), pp. 51–67, 2016.
- [22] D. Karaboga and B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm., *J Glob Optim*, 39 (2007), pp. 459–471.,
- [23] M. E. A. Elaziz, A. A. Ewees, D. Oliva, P. Duan, and S. Xiong, A Hybrid Method of Sine Cosine Algorithm and Differential Evolution for Feature Selection., 10638 (2017), pp. 145–155.

- [24] M. A. El Aziz, A. A. Ewees, A. E. Hassanien, M. Mudsh, and S. Xiong, Multi-objective Whale Optimization Algorithm for Multilevel Thresholding Segmentation., *Adv. Soft Comput. Mach. Learn. Image Process.*, 730 (2018), pp. 23–39.
- [25] A. A. Ewees and M. Abd, Engineering Applications of Artificial Intelligence Performance analysis of Chaotic Multi-Verse Harris Hawks Optimization : A case study on solving engineering problems, *Eng. Appl. Artif. Intell.*, 88 (2020), p. 103370.
- [26] J. S. and X. J. Dunwei Gong, Evolutionary algorithms with preference polyhedron for interval multi-objective optimization problems, *Inf. Sci. (Ny)*, 233 (2013), pp. 141–161.
- [27] J. Sun, Z. Miao, D. Gong, X. Zeng, J. Li, and G. Wang, Interval Multiobjective Optimization With Memetic Algorithms, *IEEE Trans. Cybern.*, 50 (2020), no. 8, pp. 3444–3457.
- [28] A. Kim, C. Wang, and S.-H. Seo, PCA-CIA Ensemble-based Feature Extraction for Bio-Key Generation, *KSII Trans. Internet Inf. Syst.*, 14 (2020), no. 7, pp. 2919–2937.
- [29] J. Torres-Jiménez and J. Pavón, Applications of metaheuristics in real-life problems., *Prog Artif Intell*, 4 (2014), pp. 175–176,.
- [30] F. V. Fomin and P. Kaski, Exact exponential algorithms., *Commun. ACM*, 56 (2013), no. 3.
- [31] E. Cicek and K. Ozturk, Optimizing the artificial neural network parameters using a biased random key genetic algorithm for time series forecasting,” *Appl. Soft Comput.*, 102 (2021), p. 107091.
- [32] Q. H. Doan, T. Le, and D.-K. Thai, Optimization strategies of neural networks for impact damage classification of RC panels in a small dataset,” *Appl. Soft Comput.*, 102 (2021), p. 107100.
- [33] J. M.S., A. A., Q. U., R. M.T., and A. K., A Novel Approach for Ensemble Feature Selection Using Clustering with Automatic Threshold,” *Int. Congr. Telemat. Comput. Springer, Cham.*, 1280 (2020), pp. 390–401.
- [34] G. Wang, Y. Wang, S. Dong, G. Huang, and Q. Sun, Optimization Methods for Power Allocation and Interference Coordination Simultaneously with MIMO and Full Duplex for Multi-Robot Networks. , *KSII Trans. Internet Inf. Syst.*, 15 (2021), no. 1, pp. 216–239.
- [35] X. Song, L. Dong, X. Huang, L. Qin, and X. Han, Energy-efficient Power Allocation based on worst-case performance optimization under channel uncertainties,” *KSII Trans. Internet Inf. Syst.*, 14 (2020), no. 11, pp. 4595–4610.
- [36] Z. Jin, C. Zhang, G. Zhao, Y. Jin, and L. Zhang, A Context-aware Task Offloading Scheme in Collaborative Vehicular Edge Computing Systems,” *KSII Trans. Internet Inf. Syst.*, 15 (2021), no. 2, pp. 383–403.
- [37] Z. Kong, D. Wu, X. Jin, S. Cen, and F. Dong, Improved AP Deployment Optimization Scheme Based on Multi-objective Particle Swarm Optimization Algorithm,” *KSII Trans. Internet Inf. Syst.*, 15 (2021), no. 4, pp. 1568–1589.
- [38] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, Multi-Verse Optimizer: a nature-inspired algorithm for global optimization,” *Neural Comput. Appl.*, 27 (2016), no. 2, pp. 495–513.
- [39] E. Hosseini, A. . Sadiq, and K. . Ghafoor, Volcano eruption algorithm for solving optimization problems., *Neural Comput Applic*, 33 (2021), pp. 2321–2337.
- [40] D. Gong, H. Yuyan, and J. Sun, A novel hybrid multi-objective artificial bee colony algorithm for blocking lot-streaming flow shop scheduling problems,” *Knowledge-Based Syst.*, 148 (2018), pp. 115–130.
- [41] D. H. Wolpert and W. G. Macready, No free lunch theorems for optimization., *IEEE Trans. Evol. Comput.*, 1 (1997), no. 1, pp. 67–82.
- [42] G. . Sayed, A. Darwish, and A. E. Hassanien, A New Chaotic Whale Optimization Algorithm for Features Selection., *J Classif*, 35 (2018), pp. 300–344.
- [43] Y. Zheng *et al.*, A Novel Hybrid Algorithm for Feature Selection Based on Whale Optimization Algorithm, *IEEE Access*, 7 (2019), pp. 14908–14923.
- [44] D. . Feldman, Chaos and Fractals: An Elementary Introduction., *Oxford Univ. Press. Oxford*, 2012.
- [45] D. Yang, G. Li, and G. Cheng, On the efficiency of chaos optimization algorithms for global optimization, *Chaos, Solitons & Fractals*, 34 (2007), no. 4, pp. 1366–1375.
- [46] A. A. Ewees, M. A. El Aziz, and A. E. Hassanien, Chaotic multi-verse optimizer-based feature selection., *Neural Comput Applic*, 31 (2019), pp. 991–1006.
- [47] G. Wang, L. Guo, A. H. Gandomi, G.-S. Hao, and H. Wang, Chaotic Krill Herd algorithm, *Inf. Sci. (Ny)*, 274 (2014), pp. 17–34.
- [48] M. Mitić, N. Vuković, M. Petrović, and Z. Miljković, Chaotic fruit fly optimization algorithm, *Knowledge-Based Syst.*, 89 (2015), pp. 446–458.
- [49] I. Rehab Ali, O. Diego, A. A. Ewees, and L. Songfeng, Feature Selection Based on Improved Runner-Root Algorithm Using Chaotic Singer Map and Opposition-Based Learning, *Neural Inf. Process.*, pp. 156–166, 2017.

- [50] B. Ren and W. Zhong, Multi-objective optimization using chaos based pso., *Inf. Technol. J.*, pp. 1908–1916, 2011.
- [51] A. H. Gandomi, G. J. Yun, X.-S. Yang, and S. Talatahari, Chaos-enhanced accelerated particle swarm optimization, *Commun. Nonlinear Sci. Numer. Simul.*, 18 (2013), pp. 327–340.
- [52] X. Han and X. Chang, A chaotic digital secure communication based on a modified gravitational search algorithm filter, *Inf. Sci. (Ny)*, 208 (2012), pp. 14–27.
- [53] M. Wang and H. Chen, Chaotic multi-swarm whale optimizer boosted support vector machine for medical diagnosis, *Appl. Soft Comput.*, 88 (2020), p. 105946.
- [54] D. Yousri, D. Allam, and M. B. Eteiba, Chaotic whale optimizer variants for parameters estimation of the chaotic behavior in Permanent Magnet Synchronous Motor, *Appl. Soft Comput.*, 74 (2019), pp. 479–503.
- [55] H. M. Burhan, B. A. Attea, A. D. Abbood, M. N. Abbas, and M. Al-Ani, Evolutionary multi-objective set cover problem for task allocation in the Internet of Things, *Appl. Soft Comput.*, 102 (2021), p. 107097.
- [56] J. García-Nieto, J. Toutouh, and E. Alba, Automatic tuning of communication protocols for vehicular ad hoc networks using metaheuristics, *Eng. Appl. Artif. Intell.*, 23 (2010), pp. 795–805.
- [57] J. Toutouh, J. Garcia-Nieto, and E. Alba, Intelligent OLSR Routing Protocol Optimization for VANETs, *IEEE Trans. Veh. Technol.*, 61 (2012), no. 4, pp. 1884–1894.
- [58] J. Toutouh, E. Alba, and S. Nesmachnow, Fast energy-aware OLSR routing in VANETs by means of a parallel evolutionary algorithm., *Clust. Comput.*, 16 (2013), pp. 435–450.
- [59] K. . Ghafoor, L. Kong, D. B. Rawat, E. Hosseini, and A. . Sadiq, “Quality of Service Aware Routing Protocol in Software-Defined Internet of Vehicles, *IEEE Internet Things J.*, 6 (2019), no. 2, pp. 2817–2828.
- [60] X. S. Yang, Nature-inspired Metaheuristic Algorithms., *Luniver Press*, 2008.
- [61] H. Gao, Y. Zhang, S. Liang, and D. Li, A new chaotic algorithm for image encryption, *Chaos, Solitons & Fractals*, 29 (2016), no. 2, pp. 393–399.
- [62] H. M. Zawbaa, E. Emary, and C. Grosan, Feature Selection via Chaotic Ant-lion Optimization., *PLoS One*, 11 (2016), no. 3, pp. 1–21.
- [63] G. Wang, S. Deb, and A. . Gandomi, Chaotic cuckoo search., *Soft Comput.*, 20 (2016), pp. 3349–3362.
- [64] G. Wang, A. . Gandomi, and H. Alavi, A chaotic particle-swarm krill herd algorithm for global numerical optimization., *Kybernetes*, 42 (2013), no. 6, pp. 962–978.
- [65] H. Yu, Y. Yu, Y. Liu, Y. Wang, and S. Gao, Chaotic grey wolf optimization, *Int. Conf. Prog. Informatics Comput.*, pp. 103–113, 2016.
- [66] D. Oliva, A. A. Ewees, M. A. El Aziz, A. E. Hassanien, and M. Pérez-Cisneros, A Chaotic Improved Artificial Bee Colony for Parameter Estimation of Photovoltaic Cells., *Energies*, 10 (2017), no. 7, p. 865.
- [67] X. Jianzhong et al, Chaotic Dynamic Weight Grey Wolf Optimizer for Numerical Function Optimization., *J. Intell. Fuzzy Syst.*, 37 (2019), no. 2, pp. 2367–2384.
- [68] X. Yao, Y. Liu, and G. Lin, Evolutionary programming made faster., *IEEE Trans. Evol. Comput.*, 3 (1999), no. 2, pp. 82–102.
- [69] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, GSA: A Gravitational Search Algorithm, *Inf. Sci. (Ny)*, 179 (2009), no. 13, pp. 2232–2248.
- [70] Liang, Jing, B. Qu, Suganthan, and Ponnuthurai, *Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization*. 2014.
- [71] J. G. Digalakis and K. G. Margaritis, On benchmarking functions for genetic algorithms., *Int. J. Comput. Math.*, 77 (2001), no. 4, pp. 481–506.
- [72] M. Molga and C. Smutnicki, Test functions for optimization needs. Test functions for optimization needs., 2005.
- [73] X. S. Yang, Engineering Optimization: An Introduction with Metaheuristic Applications., *Wiley, Chichester*, 2010.
- [74] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, *Knowledge-Based Syst.*, 89 (2015), pp. 228–249.
- [75] J. Derrac, S. García, D. Molina, and F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, *Swarm Evol. Comput.*, 1 (2011), no. 1, pp. 3–18.
- [76] F. van den Bergh and A. P. Engelbrecht, A study of particle swarm optimization particle trajectories, *Inf. Sci. (Ny)*, 176 (2006), no. 8, pp. 937–971.