A Hybrid Approach for Solving Systems of Nonlinear Equations Using Harris Hawks Optimization and Newton’s Method

RAMI SIHWAIL1, OBADAH SAID SOLAIMAN2, KHAIRUDDIN OMAR1, KHAIRUL AKRAM ZAINOL ARIFFIN3, (Member, IEEE), MOHAMMED ALSWAITTI4, (Member, IEEE), AND ISHAK HASHIM2

1Center for Artificial Intelligence, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi, Selangor 43600, Malaysia
2Department of Mathematical Sciences, Faculty of Science and Technology, Universiti Kebangsaan Malaysia, Bangi, Selangor 43600, Malaysia
3Center for Cyber Security, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi, Selangor 43600, Malaysia
4Department of Information and Communication Technology, School of Electrical and Computer Engineering, Xiamen University Malaysia, Sepang 43900, Malaysia

Corresponding author: Khairul Akram Zainol Ariffin (k.akram@ukm.edu.my)

This work was supported in part by the Ministry of Education Malaysia under Grant FRGS/1/2018/ICT04/UKM/02/3, and in part by Universiti Kebangsaan Malaysia (UKM) under Grant GUP-2019-062 and Grant GP-2019-K005539.

ABSTRACT Systems of nonlinear equations are known as the basis for many models of engineering and data science, and their accurate solutions are very critical in achieving progress in these fields. However, solving a system with multiple nonlinear equations, usually, is not an easy task. Consequently, finding a robust and accurate solution can be a very challenging problem in complex systems. In this work, a novel hybrid method namely Newton-Harris hawks optimization (NHHO) for solving systems of nonlinear equations is proposed. The proposed NHHO combines Newton’s method, with a second-order convergence where the correct digits roughly double in every step, and the Harris hawks optimization (HHO) to enhance the search mechanism, avoid local optima, improve convergence speed, and find more accurate solutions. We tested a group of six well-known benchmark systems of nonlinear equations to evaluate the efficiency of NHHO. Further, comparisons between NHHO and other optimization algorithms, including the original HHO algorithm, Particle Swarm Optimization (PSO), Ant Lion Optimizer (ALO), Butterfly Optimization Algorithm (BOA), and Equilibrium Optimization (EO) were performed. The norm of the equation system was calculated as a fitness function to measure the optimization algorithms’ performance. A solution with less fitness value is considered a better solution. Furthermore, the experimental results confirmed the superiority of NHHO over the other optimization algorithms, in the comparisons, in different aspects, including best solution, average fitness value, and convergence speed. Accordingly, the proposed NHHO is powerful and more effective in all benchmark problems in solving systems of nonlinear equations compared to the other optimization algorithms. Finally, NHHO overcomes the limitations of Newton’s method, including selecting the initial point and divergence problems.

INDEX TERMS Hybridization, Harris hawks optimization, Newton’s method, optimization, systems of nonlinear equations.

I. INTRODUCTION Systems of nonlinear equations are the basis of many fields such as engineering models and data sciences. However, finding accurate solutions to systems of nonlinear equations is a challenging issue. Let \( F(x) \) be a nonlinear multivariable function such that \( F(x) : \mathbb{R}^n \rightarrow \mathbb{R}^n \). Finding the solution \( \alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n) \) of \( F(X) = 0 \) is one of the prevalent problems in applied sciences. This type of equation can occur directly in many sciences including biology, physics, chemistry, and many engineering branches. Besides, systems of nonlinear equations can occur indirectly in many nonlinear problems that come in the form of differential equations or integral equations. These problems can be converted into...
systems of nonlinear equations using some numerical techniques such as the finite difference method [1]. Furthermore, finding the exact solution to nonlinear problems is an intricate process. For instance, there is no algebraic formula for the general solution of a quintic polynomial (Abel’s theorem) [2]. Many equations have higher orders and may include specific hyperbolic, exponential, and trigonometric terms, or they may be entirely transcendental equations that are difficult to be solved analytically to achieve the exact solution. As a consequence, the field of finding approximate solutions to nonlinear problems has become very common.

Population-based approaches (PBAs) have been used as alternatives for finding the solution(s) of nonlinear systems of equations. We can define PBAs simply as a group activity that has arisen from social insects operating under very few laws. Usually, the popular algorithms of the population-based approaches result from animals’ lifestyles. Many population-based approaches have been proposed to solve optimization problems in the literature, for instance, Harris hawks optimization (HHO) [3], equilibrium optimization (EO) [4], particle swarm optimization (PSO) [5]–[7], Slime Mould algorithm (SMA) [8], ant lion optimization (ALO) [9], grasshopper optimization algorithm (GOA) [10], genetic algorithm (GA) [11], firefly algorithm (FA) [12], Cuckoo search algorithm (CSA) [13], artificial bee colony (ABC) [14], krill herd algorithm (KHA) [15], ant colony optimization (ACO) [16], and glowworm swarm optimization (GSO) [17]. Many other examples of optimization algorithms can be found in [18].

Nonlinear equation systems have been solved using several optimization algorithms such as FA, CSA, GA, ABC, and PSO. To boost the canonical optimization algorithms performance, some researchers modified the optimization algorithm to better suit the targeted problem for more accurate solutions. For example, Ariyaratne et al. [19] presented a modified FA to solve nonlinear systems of equations in an optimization problem, such that the continuity, differentiation, and initial assumptions were not required, and it can provide multiple root approximations simultaneously. Another approach was proposed by Zhou and Li [20] to solve systems of nonlinear equations by presenting a modified CSA variant that provides a harmonious solution without modifying the cuckoo’s evolution. Similarly, Chang [21] solved systems of nonlinear equations using a suitable modification to the GA to estimate the parameters of the problems.

By presenting the systems as a multi-objective optimization problem, and then solving it using GA, Grosan and Abraham [22] solved the complex systems of nonlinear equations. In contrast, Ren et al. [23] used efficient GA with harmonious and symmetric individuals to solve nonlinear systems of equations. Also, Jaberipour et al. [24] solved systems of nonlinear equations using a modified PSO algorithm. The modification aimed at overcoming the fundamental PSO drawbacks such as trapping at local minimums and the slow convergence. Then as well, Mo and Liu [25] solved a system of nonlinear equations by an improved PSO algorithm where the improvement incorporated the conjugate direction method (CDM) into the original algorithm. The CDM increased the algorithm efficiency in optimizing high-dimensional problems and overcame the local minima incident [26].

Other researchers went further in searching for better solution approximations to the nonlinear systems of equations by combining two PBAs. As a result, the hybrid will inherit the advantages and alleviates the weaknesses of both algorithms [27]. Some examples of hybridizing PBAs are the hybrid of the ABC and PSO [28], hybrid GA [29], [30], hybrid FA [31], hybrid PSO [32], hybrid KHA [33], hybrid ABC [34], and many others [36]–[40].

Mathematicians usually employ mathematical methods that depend on successive approximations, called iterative methods, to find an approximate solution to nonlinear systems of equations. The most widespread iterative method is the one proposed by Issaak Newton, known as Newton’s method or Newton-Raphson method. Newton’s method has been widely studied and modified to increase both the accuracy of the approximate solutions and the convergence rate to the desired solution [40]–[42].

Most of the proposed optimization algorithms in the literature deal with the systems of nonlinear problems using a single optimization algorithm, or by using a hybridized algorithm of two different algorithms. In the proposed work, we use the systems of nonlinear problems using a combination of the well-known HHO optimizer and Newton’s iterative technique to benefit from the fast convergence of Newton’s method towards the desired solution. However, few types of research were proposed based on the idea of hybridizing Newton’s method with an optimization algorithm. Kart et al. [43] presented a hybrid technique that combines GA and Newton’s method for solving systems of nonlinear equations. The GA was used to find proper initial solutions which will be later fed to Newton’s method. Besides, another hybrid algorithm that combines the GA to Powell algorithm together with Newton’s method was proposed by Luo et al. [44] for solving systems of nonlinear equations. Finally, Luo et al. [45] presented a hybrid scheme for solving nonlinear systems using the quasi-Newton method and chaos optimization. However, all of these works have focused on a specific issue or a special type of problem and did not apply their proposed algorithms to arbitrary nonlinear systems of equations.

It is worth mentioning that optimization algorithms model natural phenomena where the local search mechanism in an optimization algorithm is critical to obtain better solutions. However, there is no guarantee that the search always converges to the optimal solution in the search space especially if the search space contains many candidate solutions. Further, in the case of solving nonlinear systems, optimization algorithms attempt to avoid premature convergence that can lead to local optima [18]. However, obtaining more accurate solutions in less time, or iterations, is considered an advantage for the optimization algorithm in solving nonlinear equations.
As mentioned earlier, hybridized algorithms are created by combining two optimization algorithms to overcome the demerits of one algorithm and taking advantage of the merits of the other. However, most of the hybridized algorithms still suffer from premature convergence depending on the original optimization algorithms mechanism [46]. The key point is to find a good match between the selected algorithms while creating the hybrid to obtain a more efficient variant. The following paragraph highlights the selected hybrid poles. The HHO and Newton’s method, coupled with reasons for this candidate.

On one hand, Harris Hawk optimization (HHO) is a fast, powerful, and efficient PBA. The algorithm imitates the style of Harris Hawk birds in looking and pursuing the prey in nature. A rabbit symbolizes the prey in the algorithm, which is treated as the best solution [46]. HHO algorithm has gone beyond other well-known algorithms, such as GOA, PSO, EO, ALO, and BOA [3]. However, HHO has some constraints as most optimization algorithms. Firstly, Harris hawks’ solutions may end up at local optima instead of the optimal solution. Besides, to specify the type of search that the algorithm will follow, HHO depends on the rabbit energy. The energy of the rabbit begins at approximately 2 and progressively decreases to 0 through iterations. Therefore, when the energy of the rabbit begins at approximately 2 and progressively decreases to 0 through iterations. Then, when the energy of the rabbit is greater than 1 ($E ≥ 1$) in the first half of iterations, the global search is performed. However, when the energy of the rabbit is less than 1 in the second half of iterations, HHO does not conduct a global search, although the area might not be the optimal one [47]. Hence, instead of converging to a global solution, HHO may converge to a local solution which causes a premature convergence [48].

Several researchers have modified the original HHO to accelerate the global search phase and to prevent it from falling into local search space. Dhwawale and Kamboj [49] proposes a hybrid meta-heuristic optimization technique that combines the benefits of HHO and Improved Grey wolf optimization (IGWO) to develop a more efficient method for reaching an exploration to the exploitation phase. In addition, Kamboj et al. [50] developed a hybrid variant of Harris Hawks optimizer using Sine–Cosine algorithm. Further, an improved version of the HHO algorithm that combines HHO with Canis lupus inspire grey wolf optimizer (hHHO-GWO) has been proposed by Nandi and Kamboj [51] to find the solution to various optimization problems. Recently, Krishna et al. [52] proposed a hybrid Harris hawks pattern search algorithm (hHHO-PS). In their next work, Nandi and Kamboj [53] hybridized the HHO with sine–cosine algorithm (SCA) using memetic algorithm approach to solve the photovoltaic constrained UCP of an electric power system. Other works have been proposed by improving different optimization algorithms, including SMA algorithm, to solve different numerical and engineering design challenges such as in [54] and [55].

On the other hand, Newton’s method is limited to initial point selection and divergence problems caused by improper guesses of the initial points. In other words, it performs a narrow and local search that entirely depends on the initial given values as a start with the incapability of performing a global search.

The aforementioned has motivated the proposal of an HHO hybrid for solving nonlinear equations systems. To improve the search mechanism of the HHO, Newton’s method is applied within the algorithm as a local search assistant. Newton’s iterations enhance the distribution of the initial solutions in the search space. In comparison to the random distribution used for the original HHO, Newton’s technique improves the algorithm’s computational accuracy and speeds up its convergence rate.

Therefore, this work is intended to improve the accuracy of the solution of nonlinear equations systems by combining HHO Newton’s method, as a search assistant, and HHO. The key contributions to this paper are summarized as follows:

1) To propose an improved hybrid HHO variant, namely Newton Harris hawks optimization (NHHO), that relies on Newton’s method to improve the local search in HHO and accelerate its convergence rate.
2) To apply the proposed NHHO algorithm for solving nonlinear systems of equations.
3) To evaluate the efficiency of the proposed NHHO with several well-known optimization algorithms, including HHO, PSO, EO, ALO, and BOA in terms of accuracy, fitness value, and satisfying the convergence criterion.

The rest of the work is arranged as follows: The state-of-the-art Newton’s method and HHO algorithm are introduced in Section II. Section III describes the details of the proposed NHHO. The details of the experiments, the benchmark systems, and the results are presented in Section IV. A discussion and analysis of the results are presented in Section V. Eventually, Section VI states the conclusion of the paper.

II. STATE-OF-THE-ART
A. NEWTON’S METHOD
The method which aims to find an approximate solution using successive approximations is called an iterative method. In general, iterative methods do not attain the exact solutions; because of that, researchers usually select a tolerance limit for the solutions obtained by the iterative methods, which is the difference between the exact and the approximate solutions. The most widespread iterative method is the one proposed by Isaac Newton, known as Newton’s method or Newton-Raphson method. The method is given by

$$X_{n+1} = X_n - \frac{F(X_n)}{F'(X_n)}, \quad (1)$$

where $F'(X_n)$ is the Jacobian of $F(X)$. Newton’s method is of the second order of convergence and can be applied easily to a wide range of nonlinear algebraic problems. For this reason, mathematical programs such as Mathematica and MATLAB use built-in functions based on Newton’s method to find the roots of nonlinear equations. The principle of Newton’s method is that continuous and differentiable functions can...
be approximated by a straight tangent line to them. That is, starting by initial solution $x_0$, the approximation $x_1$ is the $x$-intercept of the tangent line to the graph of $f$ at $(x_0, f(x_0))$. In the same manner, we can find the next approximation $x_2$ by finding the $x$-intercept of the tangent line to the graph of $(x_1, f(x_1))$ and so on, as illustrated in Figure 1 [56].

Newton’s method has been widely studied and modified to increase the accuracy of the approximate solutions of the nonlinear problems and increase the order of convergence that reflects consecutively on the speed of obtaining the desired solution [40]–[42].

**B. HARRIS HAWKS OPTIMIZATION (HHO)**

Harris hawks optimization (HHO) is a recent optimization algorithm introduced by Heidari et al. in 2019. HHO is considered as a meta-heuristic, population-based, and nature-inspired algorithm that imitates the Harris hawks' style in chasing rabbits. The hawks' natural strategies are applied in the algorithm, such as exploring the area looking for prey, preaching, surprise pounce, and predation strategies. As meta-heuristic algorithms, HHO is composed of exploration and exploitation phases for searching on the prey. More precisely, HHO consists of two exploration phases that make the hawks explore different areas in search of the rabbit and four stages of exploitation in which the hawks search for the rabbit in the neighborhood, as shown in Figure 2.

1) **EXPLORATION PHASE**

In the exploration phase, the Harris hawks search for the rabbit, representing the best location in the algorithm. The Harris hawks have sharp eyes that help them to monitor the area to find the rabbit. Practically, each Harris hawk is a candidate solution, and its fitness value is calculated based on its location from the rabbit. However, it is not easy sometimes to detect the prey. In this case, the Harris hawks wait on a high tree for a while hoping to observe the prey based on the following equation:

$$X(t + 1) = \begin{cases} X_{\text{rand}}(t) - r_1 |X_{\text{rand}}(t) - 2r_2X(t)| & q \geq 0.5 \\ (X_{\text{rabbit}}(t) - X_m(t)) - r_3(LB + r_4(UB - LB)) & q \leq 0.5, \end{cases}$$

(2)

where $X(t + 1)$ represents the position of the Harris hawk in the next iteration, $X_{\text{rabbit}}(t)$ is the location of the rabbit, and $X(t)$ is the current location of the hawk. The average of the current Harris hawks locations is represented by $X_m(t)$. While the variables $r_1, r_2, r_3, r_4$, and $q$ are random variables between [0, 1], $LB$ and $UB$ refer to the lower and upper limits of the variables. To calculate the average locations of the Harris hawks $X_m(t)$, HHO uses a simple way as in the following equation:

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

(3)

where $N$ represents the maximum number of Harris Hawks (solutions), and $X_i(t)$ is the location of the Harris hawks in iteration $t$.

2) **THE TRANSITION FROM EXPLORATION TO EXPLOITATION**

Indicating the right balance to shift from the exploration and exploitation phases is critical to the algorithm’s performance. HHO assumes that the rabbit’s escaping energy ($E$) is reduced while escaping from the hawks. Therefore, it relies on the value of ($E$) to perform the transitions between the two phases as shown in the following equation:

$$E = 2E_0 \left(1 - \frac{t}{T}\right),$$

(4)

where $E$ refers to the rabbit’s escaping energy, $E_0$ is random initial energy between (−1, 1), and $T$ is the maximum number
of iterations. In HHO, the exploration phase occurs when the value of $|E| \geq 1$, and the exploitation phase when $|E| < 1$, as shown in Figure 3.

3) EXPLOITATION PHASE
In the exploitation phase, the hawks identify the rabbit location in the previous stage and are ready to attack. However, the rabbit always attempts to escape from the attack. Therefore, the four exploitation phases are designed based on the absolute value of rabbit’s escaping energy $|E|$ and the probability of escaping ($r$) as illustrated in Figure 3. A summary of each attack strategy is introduced in the following.

4) SOFT BESIEGE
In the soft besiege, the rabbit has some energy with a high escaping probability as $|E| > 0.5$ and $|r| > 0.5$. In this case, the hawk softly encircles around the hawks as in the following equation:

$$X (t + 1) = \Delta X (t) - E |JX_{rabbit} - X (t)|,$$
$$\Delta X (t) = X_{rabbit} - X (t),$$

where $\Delta X (t)$ refers to the difference between the rabbit location and hawk locations in iteration $t$, and $J$ is the strength of the random rabbit jump with $J = 2(1-r_5)$ as $r_5$ is a random variable in the interval $[0, 1]$. 

5) HARD BESIEGE
In the hard besiege, the rabbit has lost most of its energy, but it still has some escaping probability as $|E| < 0.5$ and $|r| \geq 0.5$. Consequently, the Harris hawk hardly encircles around the rabbit to prepare for the final surprise pounce using the following mathematical equation:

$$X (t + 1) = X_{rabbit} (t) - E |\Delta X (t)|,$$

6) SOFT BESIEGE WITH PROGRESSIVE RAPID DIVES
In this situation with $|E| \geq 0.5$ and $|r| < 0.5$, the rabbit can still escape an attack. Accordingly, the hawk performs some dives and smart zigzag movements around the rabbit. First, the hawk tries to approach the rabbit based on the following equation:

$$Y = X_{rabbit} (t) - E |JX_{rabbit} (t) - X (t)|.$$

Second, a decision is made based on the last dive and the expected result to dive again or not. If not, the hawk begins to perform smart and irregular dive based on Levy flight concept ($LF$) as in the following equation:

$$Z = Y + S \times LF (dim),$$

where $dim$ refers to the dimension of the optimization problem and $S$ is a vector randomly generated of size $1 \times dim$. The $LF$ function is calculated as:

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

where $u$ and $v$ are random variables between $[0, 1]$, and $\beta = 1.5$ is a constant. Therefore, updating the locations of Harris hawks in soft besiege can be formulated as follows:

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

where $F$ represents the calculated fitness value for $Y$ and $Z$. 

7) HARD BESIEGE PROGRESSIVE RAPID DIVES
In the case of $|E| < 0.5$ and $|r| < 0.5$, the rabbit has low energy and less chance to escape. The hawk attempt to get closer to the rabbit by performing rapid dives before performing the surprise pounce as formulated in the following equation:

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

where

$$Y = X_{rabbit} (t) - E |JX_{rabbit} (t) - X_m (t)|,$$
$$Z = Y + S \times LF (dim).$$

Finally, the fitness function value is calculated based on the Euclidean norm, which is also called norm-2 or square norm. For this research, the solution with less norm is more accurate than a higher norm solution. Hence, a solution with norm $= 0$ is a precise solution. This norm represents the ordinary distance from the origin to the vector $F (x) = (f_1, f_2, \ldots, f_n)$ and can be defined as:

$$\downarrow \text{Fitness} = \|F (x)\|_2 = \sqrt{f_1^2 + f_2^2 + \ldots + f_n^2}. $$

Aside from all of the advantages, the HHO, like any other stochastic population-based optimizer, has some drawbacks in terms of its original ability. For instance, When the HHO population becomes stuck in local optima for a complex problem, a proper modification is needed to speed up the algorithm’s exploration. Another shortcoming is that HHO does not always maintain the proper balance between the cores, or it does not always find genuine moments, or the transition is not always smooth, particularly when dealing with very complex attribute spaces. Most of the HHO-based research works have been pointed to these two reasons to improve the convergence speed or local optima avoidance [57].
III. NEWTON-HARRIS HAWKS ALGORITHM (NHHO)

The original HHO is a robust optimization algorithm that can be used to solve many problems. However, based on the non-free lunch theorem (NFL) [58], no algorithm is perfect for solving all problems. The proposed NHHO aims to improve the performance of HHO in solving systems of nonlinear equations by adopting Newton’s method. Newton’s method in NHHO works as a local search to enhance the search mechanism in HHO as illustrated in Figure 4.

Newton’s method is applied to the rabbit location in every iteration and then the fitness value of the rabbit location is compared to the fitness of the location calculated by Newton’s method, which is represented by \( (X_{n+1}) \) in Figure 4. Consequently, the rabbit location equals the location with a minimum fitness value. The pseudocode of the proposed NHHO algorithm is presented in Algorithm 1.

The algorithm shows the different stages of the HHO algorithm, including initialization, exploration, and exploitation stages. The transition from exploration to exploitation depends on the rabbit energy \( |E| \). Exploitation is performed when the value of \( |E| \geq 1 \) and one of the four exploitation phases is selected when \( |E| < 1 \). Newton’s method, in the red box, is applied at the end of each iteration. The fitness value is calculated for the new location provided by Newton’s method and compared to the \( X_{rabbit} \) location provided by HHO. Finally, the location that has better fitness is selected.

IV. EXPERIMENTS

To evaluate the proposed NHHO performance, comparisons between NHHO and other optimization algorithms on solving six systems of nonlinear equations were performed. The six systems are general problems and known benchmarks that other researchers studied. Each problem was solved using NHHO and other well-known optimization algorithms including the original HHO, PSO, ALO, BOA, and EO algorithms.

Further, to assure fair comparisons to all optimization algorithms, the comparisons were made based on the fitness value of \( F(x) \) presented in Equation (15). The same settings have been used in all experiments, such as search agents (population size) was set to 20 and maximum iteration to 100. Furthermore, it is well-known that the performance of optimization algorithms is improved by fine parameter tuning. In this work, the parameter setting of the utilized algorithms is closed based on the original work as shown in Table 1.

The optimization algorithms were implemented using MATLAB software R2020a. The experiments were performed on an Intel i5 processor PC with 2.2 GHz, 8 GB of RAM, and Windows 8 for the operating system.

To perform the experiments six problems (benchmarks), which are commonly used in the literature, are selected. See for instance the following references [18], [22], [23], [55]. The selected problems are of different dimensions.
TABLE 1. General parameter settings of optimization algorithms.

| ALGORITHM | PARAMETER | VALUE | REFERENCE |
|-----------|-----------|-------|-----------|
| HHO       | Rabbit energy E | [2,0] | [60] |
|           | \( r_1, r_2, f_1, f_2 \) | [0,1] | |
| PSO       | Acceleration-constants C1 | 1.5 | [61] |
|           | Acceleration-constants C2 | 2 | |
|           | Inertia-Weight W1 | 1 | |
|           | Inertia-Weight W2 | 0.9 | |
| ALO       | K | 500 | [8] |
| BOA       | Power exponent a | [0.1-0.3] | [62] |
|           | Modular modality c | 0.01 | |
|           | p | 0.8 | |
| EO        | \( r_1, r_2 \) | [0,1] | [4] |
| NHHO      | Newtons initial point | Rabbit location | |
|           | Rabbit energy E | [2,0] | |
|           | \( r_1, r_2, f_1, f_2 \) | [0,1] | |

and contains several types of transcendental functions, for instance: polynomials and trigonometric functions.

A. PROBLEM 1

This problem is represented by a system of two nonlinear equations as following [23]:

\[
F(X) = \begin{cases} 
  f_1(x, y) = x^2 + y^2 + x + y - 8 = 0, \\
  f_2(x, y) = xy + x + y - 5 = 0, \\
  x \in [-3.5, 2.5], \quad y \in [-3.5, 2.5] 
\end{cases} \tag{16}
\]

The best solution obtained by optimization algorithms after running the algorithms 30 times is shown in Table 2. The performance of the provided solution is measured by calculating its fitness value \( F \).

The proposed NHHO obtained the best solution with \( x = 2 \) and \( y = 1 \), which is one of the exact solutions for the equations in problem 1. The fitness value of NHHO (0 in the case of exact solution) outperformed all other algorithms. Running the experiments for 30 runs can, sometimes, give the case of exact solution) outperformed all other algorithms. Running the experiments for 30 runs can, sometimes, give the

B. PROBLEM 2

In this problem, the system is of three nonlinear algebraic equations as following [23]:

\[
F(X) = \begin{cases} 
  f_1(x, y, z) = 3x^2 + \sin(xy) + z^2 + 2 = 0, \\
  f_2(x, y, z) = 2z^3 - y^2 - z + 3 = 0, \\
  f_3(x, y, z) = \sin(2x) + \cos(yz) + y - 1 = 0, \\
  x \in [-5, 5], \quad y \in [-1, 3], \quad z \in [-5, 5] 
\end{cases} \tag{17}
\]

The best solution obtained by the optimization algorithms for problem 2 is presented in Table 3. With fitness value \( F = 2.2204E-16 \), the proposed NHHO obtained the best solution compared to the other algorithms. The solutions in the tables are rounded to 12 digits. That is why NHHO and PSO have obtained the same solution in the table but with different fitness values. For instance, the NHHO solution rounded to 17 digits is as following \( x = -0.03275902177937206 \), \( y = 1.264628719362605 \), and \( z = 1.400643891245019 \) while the solution of PSO is \( x = -0.03275902177936415 \), \( y = 1.2646287193626 \), and \( z = 1.400643891245025 \).

C. PROBLEM 3

The system in this problem consists of two nonlinear algebraic equations; such that \( f \) is not differentiable as follows [23]:

\[
F(X) = \begin{cases} 
  f_1(x, y) = x^2 - y + 1 + \frac{1}{9} |x - 1| = 0, \\
  f_2(x, y) = y^2 - x - 7 + \frac{1}{9} |y| = 0, \\
  x \in [-2, 2], \quad y \in [1, 6]. 
\end{cases} \tag{18}
\]
TABLE 2. The best solution obtained by the compared algorithms for problem 1.

|      | NHHO   | HHO   | PSO    | ALO    | BOA    | EO     |
|------|--------|-------|--------|--------|--------|--------|
| x    | 2      | 0.9990887873709 | 1.99999999956 | 1.0000001230906 | 0.9926812384732 | 1.0000000135232 |
| y    | 1      | 2.00668866566466 | 1.0000000000038 | 1.99999979464399 | 2.001279859281 | 1.9999999901315 |
| Fitness | 0     | 1.5319E-03 | 1.0732E-11 | 4.0035E-07 | 3.0382E-02 | 2.2605E-08 |
| Std.  |        | ±0.06886 | ±1.0518 | ±0.0062 | ±0.08519 | ±0.00177 |

TABLE 3. The best solution obtained by the compared algorithms for problem 2.

|      | NHHO   | HHO   | PSO    | ALO    | BOA    | EO     |
|------|--------|-------|--------|--------|--------|--------|
| x    | -0.03275902177 | -0.04113303054 | -0.03275902177 | -0.03275896952 | -0.03935763805 | -0.03275906761 |
| y    | 1.264628719362 | 1.272157260212 | 1.264628719362 | 1.264628812475 | 1.265180044647 | 1.264628782087 |
| z    | 1.400643891245 | 1.391468702135 | 1.400643891245 | 1.400643937049 | 1.393682940116 | 1.400643846759 |
| Fitness | 2.2204E-16 | 2.0998E-02 | 1.3861E-14 | 2.9125E-07 | 1.4477E-02 | 1.4863E-07 |
| Std.  | ±5.048E-16 | ±0.08443 | ±1.126E-06 | ±0.15048 | ±0.5253 | ±0.00253 |

TABLE 4. The best solution obtained by the compared algorithms for problem 3.

|      | NHHO   | HHO   | PSO    | ALO    | BOA    | EO     |
|------|--------|-------|--------|--------|--------|--------|
| x    | 1.339811599865 | -1.072124970634 | 1.339811599865 | -1.072170989444 | 1.337963298495 | -1.072170858994 |
| y    | 2.83285196756 | 2.37979127022 | 2.832851967564 | 2.379791629447 | 2.82563938776 | 2.379791557435 |
| z    | 3.3566E-16 | 1.1353E-05 | 3.3566E-16 | 5.3002E-07 | 3.9819E-02 | 6.8182E-12 |
| Fitness | ±3.263E-16 | ±0.03768 | ±7.097E-12 | ±1.823E-05 | ±0.08552 | ±0.00011 |

Here, the best solution for problem 3 was found by the proposed NHHO and PSO with the same fitness value \( F = 3.3566E - 16 \). Besides, both of the optimization algorithms found the same solution with \( x = 1.339811599865 \) and \( y = 2.832851967564935 \). In contrast, BOA has given an approximate solution, but not as accurate as NHHO and PSO. The result of the best solution attained by running each algorithm 30 times is shown in Table 4. The graphical representation for problem 3 is illustrated in Figure 6.

D. PROBLEM 4

In this benchmark problem, there are two nonlinear algebraic equations as following [62]:

\[
\begin{align*}
F (x) &= \begin{cases} 
    f_1 (x, y) = \cos(2x) - \cos(2y) - 0.4 = 0, \\
    f_2 (x, y) = 2(y-x) + \sin(2y) - \sin(2x) - 1.2 = 0, 
\end{cases} \\
x \in [-2, 2], \quad y \in [1, 6] 
\end{align*}
\]

(19)

The best solution obtained by the optimization algorithms for problem 4 is presented in Table 5. With fitness value \( F = 1.1102E - 16 \) the proposed NHHO has achieved the best solution and outperformed the other optimization algorithms. The solutions of all algorithms, except for EO, are close to each other in the first few digits. However, the proposed NHHO has proved to be more accurate in solving nonlinear equations due to the powerful local search mechanism NHHO adopts in searching for the best solution in the search space. The graphical representation for problem 4 is illustrated in Figure 7.

FIGURE 6. The graph of the nonlinear system given by Problem 3 within the selected range.
While NHHO came first with a fitness value of $1.1102E-16$, other compared optimization algorithms with a significant difference. NHHO algorithm outperformed all of the compared algorithms for problem 4. Figure 7 illustrates the best solution obtained by the compared algorithms for problem 4.

This system of nonlinear equations is a complex system that consists of ten equations and ten variables. Usually, such a system has many solutions. In other words, the possibility is higher for optimization algorithms to be trapped in local optima. There are three different solutions in Table 5 with different fitness values. However, the hybrid NHHO can improve its solution in every iteration as it has the power of both search mechanisms; the local search of the HHO and Newton’s method.

**TABLE 5. The best solution obtained by the compared algorithms for problem 4.**

|      | NHHO   | HHO    | PSO    | ALO    | BOA    | EO     |
|------|--------|--------|--------|--------|--------|--------|
| $x$  | 0.156520069683 | 0.156520105882 | 0.156520070014 | 0.156519927799 | 0.150978929339 | 0.68023579953 |
| $y$  | 0.493376374223 | 0.4933764455054 | 0.493376374586 | 0.493376222522 | 0.487542939994 | 2.259990741424 |
| Fitness | $1.1102E-16$ | 1.2391E-07 | 4.3483E-10 | 1.8537E-07 | 7.2379E-03 | 7.9152E-11 |
| Std. | $\pm1.171E-16$ | $\pm0.01217$ | $\pm0.00055$ | $\pm1.088E-05$ | $\pm0.02256$ | $\pm0.00129$ |

**FIGURE 7.** The graph of the nonlinear system given by Problem 4 within the selected range.

**E. PROBLEM 5**
The system in this problem contains ten nonlinear algebraic equations [22]:

$$F(W) = \begin{cases} f_1 = w_1 - 0.25428722 - 0.18324757w_4w_3w_9 = 0, \\ f_2 = w_2 - 0.37842197 - 0.16275449w_1w_10w_6 = 0, \\ f_3 = w_3 - 0.27162577 - 0.16955071w_1w_2w_10 = 0, \\ f_4 = w_4 - 0.19807914 - 0.15585316w_1w_1w_6 = 0, \\ f_5 = w_5 - 0.44166728 - 0.19950920w_7w_6w_3 = 0, \\ f_6 = w_6 - 0.14654113 - 0.18922793w_8w_3w_10 = 0, \\ f_7 = w_7 - 0.42937161 - 0.21180486w_2w_5w_8 = 0, \\ f_8 = w_8 - 0.07056438 - 0.17081208w_1w_7w_6 = 0, \\ f_9 = w_9 - 0.34504906 - 0.19612740w_10w_6w_8 = 0, \\ f_{10} = w_{10} - 0.42651102 - 0.21466544w_4w_8w_10 = 0, \\ -10 \leq w_1, w_2, \ldots, w_{10} \leq 10. \end{cases}$$

Similarly, the proposed NHHO has been approved to be efficient in solving a complex set of nonlinear equations as shown in Table 6. NHHO algorithm outperformed all of the other compared optimization algorithms with a significant difference. While NHNO came first with a fitness value of $5.2147E-17$, PSO achieved a fitness value of $6.1466E-06$. The original HHO came last with fitness equals $2.7272E-01$.

The best solution obtained by the optimization algorithms for problem 6 is presented in Table 7. With fitness value $(F) = 6.9871E-21$, the proposed NHHO obtained the best solution compared to the other algorithms. PSO and EO came second with fitness values equal to $1.5098E - 11$ and $7.2487E - 10$, respectively. From problem 5 and problem 6, it can be noticed that the proposed NHHO is flexible and can work efficiently even in a wide interval $[-10, 10]$.

Eventually, the obtained results by the proposed hybrid NHHO on all the previous problems confirm the theoretical hypothesis discussed earlier, i.e., the hybrid will inherit the powerful merits from both algorithms. This is obvious in case the comparison has been made separately between NHHO and the HHO where the proposed hybrid surpasses the original HHO in all obtained fitness values. This is imputed to the employed local search (Newton’s method) which enhanced the hybrid capability to avoid the local optima in Problem 1.
TABLE 6. The best solution obtained by the compared algorithms for problem 5.

|   | NHHO     | HHO     | PSO     | ALO     | BOA     | EO      |
|---|----------|---------|---------|---------|---------|---------|
| w₁ | 0.257833393700 | 0.229605360542 | 0.257833029557 | 0.223623964826 | 0.300803744166 | 0.25757653401 |
| w₂ | 0.381009154602 | 0.204652945825 | 0.381079080935 | 0.354019824030 | 0.342575458374 | 0.380869284780 |
| w₃ | 0.278745017346 | 0.257927660932 | 0.278745916690 | 0.289983409078 | 0.246966364042 | 0.278668734628 |
| w₄ | 0.200668964225 | 0.256923275135 | 0.200668617541 | 0.196050351349 | 0.277439012172 | 0.200383571935 |
| w₅ | 0.445251424841 | 0.466570438348 | 0.445251146273 | 0.442343486362 | 0.405494080044 | 0.44540193541 |
| w₆ | 0.149183991969 | 0.204470872060 | 0.149177899419 | 0.104103282726 | 0.173143141941 | 0.149530461472 |
| w₇ | 0.432009698983 | 0.435526895328 | 0.432010282483 | 0.353640262400 | 0.402143545537 | 0.432272963087 |
| w₈ | 0.073402777776 | 0.246845270769 | 0.073402588107 | 0.087928198968 | 0.045624012773 | 0.073976538341 |
| w₉ | 0.345966826875 | 0.344230077978 | 0.345966494281 | 0.343964544316 | 0.195838179757 | 0.345197879244 |
| w₁₀ | 0.427326275993 | 0.350774420086 | 0.427326633304 | 0.424636187969 | 0.337222641638 | 0.427216959402 |
| Fitness | 5.2147E-17 | 2.7272E-01 | 6.1466E-06 | 1.2261E-01 | 2.1106E-01 | 1.1641E-03 |
| Std.   | ±4.1355E-16 | ±0.06307 | ±0.01223 | ±1.5294 | ±0.17463 | ±0.09744 |

TABLE 7. The best solution obtained by the compared algorithms for problem 6.

|   | NHHO     | HHO     | PSO     | ALO     | BOA     | EO      |
|---|----------|---------|---------|---------|---------|---------|
| w₁ | 1.84926832E-07 | 1.00568654E-22 | 2.66120199E-07 | 1.26336053E-05 | 1.24809688E-05 | 1.83495888E-05 |
| w₂ | 1.57940299E-07 | 1.00568654E-22 | 7.18179256E-15 | 3.20191220E-06 | 1.12E-22 | 4.74379143E-07 |
| w₃ | 1.38643718E-05 | 1.00568654E-22 | 2.08119459E-05 | 3.00021464E-05 | 1.67949832E-05 | 2.93489584E-05 |
| w₄ | 6.39244122E-11 | 7.01514317E-06 | 3.78200878E-06 | 9.50207569E-06 | 1.74539283E-06 | 8.46983256E-09 |
| w₅ | 6.65272882E-21 | 1.00568654E-22 | 1.00000000E-22 | 1.12E-22 | 1.12E-22 | 2.01664290E-22 |
| w₆ | 2.47734088E-06 | 1.05299247E-12 | 6.01934856E-08 | 4.59487181E-10 | 4.55411493E-06 | 4.76289181E-06 |
| w₇ | 4.99996803E-06 | 1.00568654E-22 | 3.10899545E-06 | 2.42343707E-07 | 5.01627780E-06 | 4.99582081E-06 |
| w₈ | 1.71356281E-05 | 2.04652817E-05 | 1.01880516E-05 | 9.31857154E-07 | 9.28614649E-06 | 1.65081707E-06 |
| w₉ | 4.71512129E-07 | 1.35341339E-05 | 9.21205031E-06 | 6.27499712E-06 | 1.12E-22 | 3.23650366E-22 |
| w₁₀ | 2.20793289E-06 | 1.00568654E-22 | 3.33780975E-07 | 2.55455909E-07 | 1.12E-22 | 1.28495633E-22 |
| Fitness | 6.9871E-21 | 1.2016E-05 | 1.5098E-11 | 7.4518E-08 | 6.1713E-06 | 7.2487E-10 |
| Std.   | ±0.04470 | ±0.02190 | ±7.37220 | ±0.30929 | ±0.41590 |

Based on the results, NHHO has ranked first, outperforming all the other compared algorithms in an average of 30 runs. First, there is a significant difference between the average fitness of the proposed NHHO and the original HHO. Accordingly, NHHO has more flexibility to find the best solution and avoid falling into local optima. Second, comparing NHHO with PSO, ALO, BOA, and EO optimization algorithms, the significant improvement can be seen clearly in all problems especially in problem 1 and problem 5. Further, it is crucial to maintain consistency in the 30 runs. In other words, to achieve consistency the average of solutions has to be close, if not the same, to the best solution. The consistency is maintained in five problems in NHHO except for problem 6.

On the other hand, convergence speed is another critical issue in optimization algorithms. The proposed NHHO improves the convergence speed of HHO by avoiding premature convergence and having the right balance between the

V. RESULTS AND ANALYSIS
To analyze the efficiency and the repeatability of the proposed NHHO algorithm results, the average of the 30 runs should be considered and not only the best solution, which represents the best case. Therefore, the average fitness values for the previous six problems are calculated. The average fitness results are presented in Table 8.

(since the fitness is 0) and significantly improve the obtained fitness values of all other algorithms in the remaining problems (which will be considered as the global optima until better solutions could be found). Furthermore, the obtained standard deviation values by the proposed hybrid indicate more stable and repeatable results in the majority of the problems. The aforementioned observations are considered as a piece of strong evidence that the proposed method has successfully avoided the entrapment at least in the local optima that all other algorithms have been in the tested problems.
exploration and exploitation phases. A comparison between the NHHO algorithm and other optimization algorithms is shown in Figure 8.

It can be noticed from the results that NHHO can reach the best solution faster than all the other optimization algorithms, which indicates the superiority of NHHO. Further, the NHHO outperformed all the optimization algorithms in all problems except for problem 3, where NHHO has almost the same fitness as PSO. However, based on Figure 8, NHHO reached the solution much faster than PSO. The average of reaching the best solution in 30 runs for NHHO is 17 iterations, while the same fitness value needed an average of 72 iterations for PSO.

The convergence of the new hybrid algorithm is due to the combination of Newton’s method, as a local search, and HHO algorithm. On one hand, Newton’s method is designed for solving nonlinear systems. On the other hand, HHO guarantees a very good selection of initial guesses for Newton’s method. Further, it is well-known that Newton’s method is considered as a second-order convergence, which means that the number of correct digits roughly doubles in every step [41]. This advantage does not exist in any optimization algorithm, even in the HHO algorithm without utilizing Newton’s method.

However, each optimization algorithm suffers from some limitations that make it difficult for the algorithm to locate the best solution. PSO has some limitations, such as poor population diversity and balancing between local optima. BOA gets stuck into local optima and it lacks exploitation of local values [48]. The limitations of ALO include the trapping of ants in antlions’ trip which prevents the algorithm from identifying the global optimum solution. In contrast, the EO algorithm does not work effectively for large-scale problems [63].

Newton’s method has been approved to be very effective when combines with HHO algorithm. Figure 9 presents a comparison between NHHO and other optimization algorithms in the six problems’ fitness value in an average of 30 runs. The y-axis in Figure 9 must be set to a logarithmic scale as the gap between the values is too big. The average fitness of NHHO is $4.2267E - 10$, while it is $0.07935$ for the original HHO, $0.048372$ for PSO, $0.33967$ for ALO, $0.167762$ for BOA, and $0.001294$ for EO.

It is worth mentioning that all calculations have been performed using variables with 16 digits of precisions. The 16 digits variable is the default for MATLAB software and it is time-saving comparing to more significant digits. However, the 16 digits may limit the calculation and affect the results in some cases. The MATLAB function ‘vpa’ can be used to increase the number of digits. Increasing the number of digits may enhance the experimental results, but it is undoubtedly a time-consuming process. A comparison of applying NHHO to the six benchmark problems with other optimization algorithms in terms of fitness value and computational time based on variables of 16, 32, and 64 digits is shown in Table 9.

The results show a significant improvement in the fitness for all tested problems. For example, a fitness of $3.357E-16$ using 16 digits to $1.504E-39$ using 32 digits and to $1.082E-71$ using 64 digits for problem 3. However, the consumed time increased dramatically from 1.4 seconds using variables of 16 digits to 8.8 seconds and 9.1 seconds, when using variables of 32 digits and 64 digits, respectively.

The standard deviation has been calculated for all problems to examine the stability of the proposed NHHO in finding the best solution in 30 runs. As can be noticed in Tables 2-6, the value of the standard deviation is zero in Table 2 and close to zero in Table 3-6. The results of NHHO outperformed the other algorithms which confirm the stability of NHHO in finding the best solutions in the benchmark. In benchmark problem 6, Table 7, NHHO came second with a standard deviation of $\pm 0.04470$ slightly higher than the original HHO that calculated $\pm 0.02190$.

However, every improvement comes with a cost. Despite the efficiency of the proposed hybrid, the overall complexity is quite higher if compared to pure algorithms. This is a common cause within hybrids where more than one algorithm complexity is added to another. While optimizing a problem solution, there should be a tradeoff between gain and loss factors. The goal of proposing such a hybrid was to enhance these three factors (best and average fitness and the convergence speed) at the expense of a slight increase in the required time and the complexity in finding the optimal solution.
This concept was inspired by Stephen Boyd [64] as the following quote “Global optimization is used for problems with a small number of variables, where computing time is not critical, and the value of finding the true global solution is very high.” Further details are provided in subsection D.

A. COMPARISON WITH NEWTON’S METHOD
To clarify the efficiency of NHHO on the accuracy of the obtained solutions, and its ability to avoid local optima compared to Newton’s method, we applied both techniques to problems 1-4. A comparison between the fitness values obtained by NHHO and Newton’s method using three arbitrary initial points is shown in Tables 10-13. For comparison purposes, we tested both schemes at the 5th, 7th, and 10th iteration (Iter.). Besides, the best solution achieved by NHHO, in 10 runs and 1200 digits variables, was selected for each problem.

Tables 10-13 show that NHHO outperformed Newton’s method in all selected problems. The main critical issue while applying Newton’s method is the choice of the
initial point because the method is very sensitive to the selected initial point. Choosing an improper initial point may affect Newton’s method by slowing its convergence (see Table 11-12), or even make the method diverge (see Table 13), and sometimes the selected initial point may cause a singularity to the Jacobian matrix of $F$, therefore, the inverse of the Jacobian does not exist. Consequently, Newton’s method cannot be applied (see Table 10).

The proposed NHHO overcomes these problems by selecting random multiple initial points, which are called search agents. NHHO tests the fitness of all search agents and the best search agent, which achieves the lowest fitness, is selected as an initial point. Accordingly, the selection of the best search agents plays an important role in overcoming the problem of choosing a proper initial point and eventually the solution’s accuracy.

The results from Tables 10-13 also confirmed the dominance of NHHO over Newton’s method. The reason for the significant improvement in the results is the ability of the hybrid NHHO to overcome the limitation of the initial
TABLE 15. The best solutions for applying NHHO to the benchmark problems with different initial intervals.

| Problem | [-10, 10] | [-100, 100] | [-500, 500] | [-1000, 1000] | [-5000, 5000] | [-10000, 10000] |
|---------|-----------|-------------|-------------|---------------|---------------|----------------|
| Problem 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Problem 2 | 2.220E-16 | 2.220E-16 | 2.220E-16 | 2.220E-16 | 2.220E-16 | 2.220E-16 |
| Problem 3 | 3.357E-16 | 3.357E-16 | 3.357E-16 | 3.357E-16 | 3.357E-16 | 3.357E-16 |
| Problem 4 | 1.110E-16 | 1.110E-16 | 1.110E-16 | 1.110E-16 | 1.110E-16 | 1.110E-16 |
| Problem 5 | 5.215E-17 | 5.215E-17 | 5.215E-17 | 5.215E-17 | 5.215E-17 | 5.215E-17 |
| Problem 6 | 6.987E-21 | 6.987E-21 | 6.987E-21 | 6.987E-21 | 6.987E-21 | 6.987E-21 |

guess in Newton’s method by relying on 20 search agents at the initial stage instead of one as Newton’s method does. In these experiments, we fixed the variable to 1200 digits, and that is the reason why the fitness results for NHHO in iteration (Iter.) 10 are equal $E = 1207$ and $E = 1208$. Accordingly, the results may be improved as we increase the number of digits as mentioned previously.

B. COMPARISON WITH RELATED WORK
In this section, NHHO is compared with the related work of El-Shorbagy [18]. The authors in the related work proposed a hybrid technique (GOA-GA) that combines grasshopper optimization algorithm with genetic algorithm to solve systems of nonlinear equations. Hybrid-GOA-GA take the merits of both GOA and GA; where GOA’s exploitability and GOA’s exploration potential are combined.

Further, this work performed the experiments on the same problems (benchmarks) used in El-Shorbagy’s work. Therefore, a comparison between the results of both algorithms for the six problems based on the best solution achieved in 30 runs was performed, and the achieved fitness values are presented in Table 14.

As noted from the results, NHHO outperformed the hybrid GOA-GA in all benchmark problems. The consistency of GOA-GA is reduced as the initial interval is expanded. For example, the fitness value increased from 5.3607E-09 to 2.7607E-04 when expanding the initial interval to the length of 10000 in problem 1. The increase in fitness value is noticed in all other problems as well, which affected the stability of the approach to be generalized to other benchmark problems. However, NHHO is more consistent as explained in the next section.

C. CONSISTENCY ON DIFFERENT INITIAL INTERVALS
It is critical to study the performance of the proposed NHHO over different initial intervals. Therefore, NHHO was run on every benchmark problem for 10 times at the following initial intervals $[-10, 10], [-100, 100], [-500, 500], [-1000, 1000], [-5000, 5000], \text{and} [-10000, 10000]$. Table 15 shows the best fitness achieved for all problems. It is very clear, based on the results, that NHHO shows high consistency by achieving the same fitness values in all initial intervals.

The hybrid approach has a very strong search mechanism that applies Newton’s method to search for the best solution even in a large search space such as in applying the initial interval of $[-10000, 10000]$. Although the initial intervals have widely expanded, NHHO managed to get the same best solutions mentioned in Tables 2-7.

However, the number of times NHHO reached the best solutions were not the same; while the best solutions were achieved almost in all of the 10 runs in the initial interval of $[-10, 10]$, they were only achieved few times in the interval $[-10000, 10000]$. Accordingly, the proposed NHHO can avoid trapping into local optima even in large search spaces and that approves the superiority of NHHO.

D. COMPUTATIONAL COMPLEXITY
The time complexity of NHHO relies on the following three factors: the initial phase, updating the Harris hawks’ positions, and applying Newton’s method. The initialization process complexity equals $O(N)$, where $N$ refers to the number of Harris hawks. The complexity of updating process, including updating the positions of the hawks and finding the best solution is $O(T \times N) + O(T \times N \times C)$, where $T$ represents the maximum number of iterations, and $C$ is the complexity of the benchmark equations. Further, the complexity of applying Newton’s method is calculated as $O(T \times W)$, where $W$ indicates Newton’s method computational time. Therefore, the total computational complexity of the proposed NHHO is $O(N \times (T + TC + 1) + TW)$.

The complexity of NHHO is calculated based on adding the complexity of HHO to the complexity of Newton’s method. It is worth mentioning here that the complexity of Newton’s method is considered too high compared to the complexity of optimization algorithms. Therefore, it is expected that combining Newton’s method with any optimization algorithm will increase its complexity by $O(T \times W)$ as Newton’s method performs an additional local search. The computational times of HHO, Newton’s method, and NHHO based on 100 iterations ($T = 100$) are shown in Table 16. However, to have a fair comparison, the experiments on NHHO performed on a stopping criterion that the achieved fitness value is less than $1 \times E-15$ ($\epsilon \leq 1 \times E-15$) as it is well known that Newton’s method can reach a significant fitness value with few iterations.
TABLE 16. A comparison of average time in seconds between HHO, Newton’s method and NHHO.

| Problem | HHO T=100 | Newton T=100 | NHHO T=100 | NHHO $\varepsilon \leq 1 \times 10^{-15}$ |
|---------|-----------|--------------|------------|---------------------------------|
| Problem 1 | 0.059 | 2.533 | 2.570 | 0.430 |
| Problem 2 | 0.078 | 4.814 | 4.926 | 0.693 |
| Problem 3 | 0.048 | 2.592 | 2.653 | 0.437 |
| Problem 4 | 0.076 | 3.253 | 3.325 | 0.419 |
| Problem 5 | 0.063 | 11.937 | 12.439 | 2.565 |
| Problem 6 | 0.046 | 10.660 | 10.863 | 2.373 |

Note that an N by N system of linear equations must be solved at each iteration. Thus, such derivative-based methods can be time-consuming because any computation of the Jacobian needs $n^2$ evaluations of scalar functions.

Combining HHO with Newton’s method to develop NHHO did not add significant computational time compared to Newton’s method as can be noticed from Table 16. However, NHHO can overcome the limitations of Newton’s method, as discussed previously, such as selecting the initial points and divergence problems. Therefore, it is more effective in solving nonlinear equation systems.

VI. CONCLUSION

In this work, a hybrid between a powerful optimization algorithm (HHO) and an iterative approximation method (Newton’s method) was proposed to solve systems of nonlinear equations. The role of the HHO was to guarantee a good selection of initial guesses for Newton’s method to widen its applicability to different problems where Newton’s method was employed as a local search mechanism with the advantage of being a second-order convergence method.

In the experiments, we utilized six well-known problems that consist of complex systems of nonlinear equations to evaluate the performance of NHHO. Besides, NHHO was compared to well-known optimization algorithms such as the original HHO, PSO, EO, ALO, and BOA. The experiments were evaluated based on each optimization algorithm’s fitness value, which was calculated based on the norm of the equations. The results have confirmed the superiority of NHHO over other algorithms in solving systems of nonlinear equations in terms of the best solution, fitness value, and convergence speed. Besides, the results of running the algorithms 30 times confirmed the consistency of NHHO.

Moreover, the nonlinear system’s solution can be significantly improved by increasing the number of digits for the variables used by NHHO but it is time-consuming. The consistency of NHHO has been also approved on different initial intervals. Finally, NHHO overcomes the limitations of Newton’s method including initial point selection and divergence problems.

The scope of this manuscript is to evaluate the ability of the proposed NHHO algorithm in solving systems of nonlinear equations and hypothetically, all the solved problems were considered as global optimization problems since no studies have provided a clear categorization of them. The performance of the proposed algorithm in solving standard optimization benchmarks may be further investigated in the future.

The motivation of this work is to solve systems of nonlinear equations. However, the effect of NHHO on other types of problems, such as unimodal and multimodal problems will be considered for future work.

We believe that the local search of Newton’s method can be applied to other optimization algorithms to improve their search mechanisms for future work. Besides, NHHO can be applied to real-life problems such as data science and engineering problems.

REFERENCES

[1] A. Burden, R. L. Burden, and J. D. Faires, Numerical Analysis, 10th ed. Boston, MA, USA: Cengage Learning, 2016.
[2] V. B. Alekseev, Abel’s Theorem in Problems and Solutions. Dordrecht, The Netherlands: Kluwer, 2004.
[3] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, “Harris hawks optimization: Algorithm and applications,” Future Gener. Comput. Syst., vol. 97, pp. 849–872, Aug. 2019, doi: 10.1016/j.future.2019.02.028.
[4] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, “Equilibrium optimizer: A novel optimization algorithm,” Knowl.-Based.Syst., vol. 191, Mar. 2020, no. 105190, doi: 10.1016/j.knosys.2019.105190.
[5] M. A. El-Shorbagy and A. E. Hassanien, “Particle swarm optimization from theory to applications,” Int. J. Rough Sets Data Anal., vol. 5, no. 2, pp. 1–24, Apr. 2018, doi: 10.4018/ijrsda.2018040101.
[6] M. A. E. Shorbagy and A. A. Mousa, “Chaotic particle swarm optimization for imprecise combined economic and emission dispatch problem,” Rev. Inf. Eng. Appl., vol. 4, no. 1, pp. 20–35, 2017, doi: 10.18488/journal.79.2017.41.20.35.
[7] A. A. A. Mousa and M. A. El-Shorbagy, “Enhanced particle swarm optimization based local search for reactive power compensation problem,” Appl. Math., vol. 3, no. 10, pp. 1276–1284, 2012, doi: 10.4236/am.2012.330184.
[8] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, “Simile mould algorithm: A new method for stochastic optimization,” Future Gener. Comput. Syst., vol. 111, pp. 300–323, Oct. 2020, doi: 10.1016/j.future.2020.03.055.
[9] S. Mirjalili, “The ant lion optimizer,” Adv. Eng. Softw., vol. 83, pp. 80–98, May 2015, doi: 10.1016/j.advengsoft.2015.01.010.
[10] S. Saremi, S. Mirjalili, and A. Lewis, “Grasshopper optimisation algorithm: Theory and application,” Adv. Eng. Softw., vol. 105, pp. 30–47, Mar. 2017, doi: 10.1016/j.advengsoft.2017.01.004.
[11] J. H. Holland, “Genetic algorithms,” Sci. Amer., vol. 267, no. 1, pp. 66–72, Jul. 1992, doi: 10.1038/scientificamerican0792-66.
[12] M. A. E. Shorbagy and A. A. Mousa, “Chaotic particle swarm optimization based local search for reactive power compensation problem,” Appl. Math., vol. 3, no. 10, pp. 1276–1284, 2012, doi: 10.4236/am.2012.330184.
[13] M. A. Elshishy, D. A. Hammad, and M. A. El-Shorbagy, “Solving interval quadratic programming problems by using the numerical method and swarm algorithms,” Complexity, vol. 2020, pp. 1–11, Sep. 2020, doi: 10.1155/2020/6105952.
[14] M. Shehab, A. T. Khader, M. Laouachedi, and O. A. Alomari, “Hybridizing cuckoo search algorithm with bat algorithm for global numerical optimization,” J. Supercomput., vol. 75, no. 5, pp. 2395–2422, May 2019, doi: 10.1007/s11227-018-2625-x.
[15] D. Karaboga, “An idea based on honey bee swarm for numerical optimization,” Dept. Eng., Erciyes Univ., Kayseri, Turkey, Tech. Rep., 2005, pp. 1–10, vol. 200. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.714.4934.
[16] A. L. Bolaji, M. A. Al-Betar, M. A. Awadallah, A. T. Khader, and L. M. Abualigah, “A comprehensive review: Krill herd algorithm (KH) and its applications,” Appl Soft Comput., vol. 49, pp. 437–446, Dec. 2016, doi: 10.1016/j.asoc.2016.08.041.
[17] M. Dorigo and T. Stützle, Ant Colony Optimization. Cambridge, MA, USA: MIT Press, 2004.
[18] M. Marinakis and Y. Marinakis, “A glowworm swarm optimization algorithm for the vehicle routing problem with stochastic demands,” Expert Syst. Appl., vol. 46, pp. 145–163, Mar. 2016, doi: 10.1016/j.eswa.2015.10.012.
M. A. El-Shorbagy and A. M. El-Refaey, “Hybridization of grasshopper optimization algorithm with genetic algorithm for solving system of nonlinear equations,” IEEE Access, vol. 8, pp. 220944–220961, 2020, doi: 10.1109/ACCESS.2020.3043029.

M. A. A. Ariyaratne, T. G. I. Fernando, and S. Weerakoon, “Solving systems of nonlinear equations using a modified firefly algorithm (MFA),” Swarm Evol. Comput., vol. 48, pp. 72–92, Aug. 2019, doi: 10.1016/j.swevo.2019.03.010.

R. H. Zhou and Y. G. Li, “An improved cuckoo search algorithm for solving nonlinear equation group,” Appl. Mech. Mater., vol. 655–653, pp. 2121–2124, Sep. 2014, doi: 10.4028/www.scientific.net/AMM.651-653.2121.

W.-D. Chang, “An improved real-coded genetic algorithm for parameters estimation of nonlinear systems,” Mech. Syst. Signal Process., vol. 20, no. 1, pp. 236–246, Jan. 2006, doi: 10.1016/j.ymssp.2005.05.007.

C. Grosan and A. Abraham, “A new approach for solving nonlinear equations system,” IEEE Trans. Syst., Man, Cybern. Syst., vol. 38, no. 3, pp. 698–714, May 2008, doi: 10.1109/TSMCA.2008.918599.

H. Ren, L. Wu, W. Bi, and I. K. Argyros, “Solving nonlinear equations and eighth-order of convergence derivative-free modifications of king’s method,” Int. J. King Saud Univ.-Sci., vol. 31, no. 4, pp. 1499–1504, Oct. 2019, doi: 10.1016/j.ijksus.2018.12.001.

O. S. Solaiman and I. Hashim, “An iterative scheme of arbitrary odd order and its basins of attraction for nonlinear systems,” Comput. Math. Continua, vol. 66, no. 2, pp. 1427–1444, 2021, doi: 10.32604/cm.2020.012610.

O. S. Solaiman and I. Hashim, “Optimal eighth-order solver for nonlinear equations with applications in chemical engineering,” Intell. Autom. Soft Comput., vol. 27, no. 2, pp. 379–390, 2021, doi: 10.32604/iasc.2021.015285.

C. L. Karr, B. Weck, and L. M. Freeman, “Solutions to systems of nonlinear equations via a genetic algorithm,” Eng. Appl. Artif. Intell., vol. 11, no. 3, pp. 369–375, Jun. 1998, doi: 10.1016/S0950-7051(97)90007-5.

Y. Z. Luo, D. C. Yuan, and G. J. Tang, “Hybrid genetic algorithm for solving systems of nonlinear equations,” Chin. J. Comput. Mech., vol. 22, no. 1, pp. 109–114, 2005.

Y.-Z. Luo, J.-G. Tang, and L.-N. Zhou, “Hybrid approach for solving systems of nonlinear equations using chaos optimization and quasi-Newton method,” Appl. Soft Comput. J., vol. 8, no. 2, pp. 1068–1073, Mar. 2008, doi: 10.1016/j.asoc.2007.05.013.

R. Sihwail, K. Omar, A. Z. Ariffin, and M. Tubishat, “Improved Harris hawks optimization using elite opposition-based learning and novel search mechanism for feature selection,” IEEE Access, vol. 8, pp. 121127–121145, 2020, doi: 10.1109/ACCESS.2020.3006473.

H. Jia, C. Lang, D. Oliva, W. Song, and X. Peng, “Dynamic Harris hawks optimization with mutation mechanism for satellite image segmentation,” Remot. Sens., vol. 11, no. 12, p. 1421, Jun. 2019, doi: 10.3390/rs11121421.

H. M. Ridha, A. A. Heidari, M. Wang, and H. Chen, “Boosted mutation-based Harris hawks optimizer for parameter identification of single-diode solar cell models,” Energy Convers. Manage., vol. 209, Apr. 2020, Art. no. 112660, doi: 10.1016/j.enconman.2020.112660.

D. Dhawale and K. V. Kumar, “hhHO-IGWO: A new hybrid Harris hawks optimizer for solving global optimization problems,” in Proc. Int. Conf. Comput. Automat. Knowl. Manage. (ICCKAM), 2020, pp. 52–57, doi: 10.1109/icckam46823.2020.9051509.

V. K. Kamboj, A. Nandi, A. Bhadoria, and S. Sehgal, “An intensify Harris hawks optimizer for numerical and engineering optimization problems,” Appl. Soft Comput. J., vol. 89, Apr. 2020, Art. no. 106018, doi: 10.1016/j.asoc.2019.106018.

A. Nandi and V. K. Kamboj, “A canis lupus inspired upward Harris hawks optimizer for nonlinear, constrained, continuous, and discrete engineering design problem,” Int. J. Numer. Methods Eng., vol. 122, no. 4, pp. 1051–1088, Feb. 2021, doi: 10.1002/nme.6573.

A. B. Krishna, S. Saxena, and V. K. Kamboj, “A novel statistical approach to numerical and multidisciplinary design optimization problems using pattern search inspired Harris hawks optimizer,” Neural Comput. Appl., vol. 33, no. 12, pp. 7031–7072, Jun. 2021.

A. Nandi and V. K. Kamboj, “A meiorated Harris hawks optimizer for combinatorial unit commitment problem with photovoltaic applications,” J. Electr. Syst. Inf. Technol., vol. 8, no. 1, pp. 1–73, Dec. 2021, doi: 10.1186/s43067-020-00026-3.

A. B. Krishna, S. Saxena, and V. K. Kamboj, “HSMA-PS: A novel memetic approach for numerical and engineering design challenges,” Eng. Comput., vol. 23, pp. 1–35, Apr. 2021.

D. Dhawale, V. K. Kamboj, and P. Anand, “An effective solution to numerical and multi-disciplinary design optimization problems using chaotic slime mold algorithm,” Eng. Comput., vol. 1–39, May 2021.

G. Strang and E. Herman, Calculus, Houston, TX, USA: Openstax, 2016.

H. M. Alabdo, D. Alabardat, L. Abualigah, and A. A. Heidari, “Harris hawks optimization: A comprehensive review of recent variants and applications,” Neural Comput. Appl., pp. 1–42, Feb. 2021, doi: 10.1007/s00521-021-05720-5.

S. P. Adam, S. A. N. Alexandropoulos, P. M. Pardalos, and M. N. Vrahatis, “No free lunch theorem: A review,” in Approximation and Optimization (Springer Optimization and Its Applications), vol. 145, Cham, Switzerland: Springer, 2019, pp. 57–92, doi: 10.1007/978-3-030-12767-1_5.
and data science and analytics. Analysis and detection, digital forensics, feature selection, feature extraction, image transforms, e.g., Trace, Fourier, and Wavelet) with applications to segmentation, feature extraction, image improvement, texture, resolution, uncertainty–Bayesian reasoning; neural networks; fuzzy logic; and fuzzy neural networks, image processing–2D and 3D, edge detection, thinning, segmentation, feature extraction, image improvement, texture, resolution, image transforms, e.g., Trace, Fourier, and Wavelet) with applications to Jawi/Arabic manuscripts and biometric authentication.

[59] T. Thaker, A. A. Heidari, M. Mafarja, J. S. Dong, and S. Mirjalili, “Binary Harris hawks optimizer for high-dimensional, low sample size feature selection,” in *Evolutionary Machine Learning Techniques*. Singapore: Springer, 2020, pp. 251–272, doi: 10.1007/978-981-32-9990-0_12.

[60] G. I. Sayed, A. Tharwat, and A. E. Hassanien, “Chaotic dragonfly algorithm: An improved metaheuristic algorithm for feature selection,” *Int. J. Speech Technol.*, vol. 49, no. 1, pp. 188–205, Jan. 2019, doi: 10.1007/s10489-018-1261-8.

[61] S. Arora and S. Singh, “Butterfly optimization algorithm: A novel approach for global optimization,” *Soft Comput.*, vol. 23, no. 3, pp. 715–734, Feb. 2019, doi: 10.1007/s00500-018-3102-4.

[62] A. Pourrajabian, R. Ebrahimi, M. Mirzaei, and M. Shams, “Applying genetic algorithms for solving nonlinear algebraic equations,” *Appl. Math. Comput.*, vol. 219, no. 24, pp. 11483–11494, Aug. 2013, doi: 10.1016/j.amc.2013.05.057.

[63] Z. M. Elgamal, N. M. Yasin, A. Q. M. Sabri, R. Sihwail, M. Tubishat, and H. Jarrah, “Improved equilibrium optimization algorithm using elite opposition-based learning and new local search strategy for feature selection in medical datasets,” *Comput. Sci. Res. Technol.*, vol. 9, no. 6, pp. 68, Jun. 2021, doi: 10.3390/computation9060068.

[64] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex Optimization*, Cambridge, U.K.: Cambridge Univ. Press, 2004.

**RAMI SIHWAIL** received the B.Sc. degree in computer science from Al Albayt University, Jordan, in 1999, the M.Sc. degree in computer science from University Putra Malaysia (UPM), in 2002, and the Ph.D. degree in computer science from Universiti Kebangsaan Malaysia (UKM), Malaysia, in 2021. In March 2010, he joined King Faisal University as a Computer Science Lecturer for nine years. He is specialized in malware analysis and live forensics. His research interests include machine learning, optimization algorithms, cybersecurity, malware analysis and detection, digital forensics, feature selection, feature extraction, and data science and analytics.

**OBADAH SAID SOLAIMAN** was born in Homs, Syria, in 1985. He received the B.Sc. and M.Sc. degrees in mathematics from the Jordan University of Science and Technology, Jordan, in February 2007 and January 2010, respectively, and the Ph.D. degree in mathematics from Universiti Kebangsaan Malaysia, Malaysia, in January 2021, under the supervision of Prof. Ishak Hashim in numerical analysis. In November 2010, he joined King Faisal University as a Mathematics Lecturer for nine years. His research interests include applied mathematics include analytical/numeric-analytic methods for nonlinear equations.

**KHAIRUDDIN OMAR** received the bachelor's and master's degrees in computer science from Universiti Kebangsaan Malaysia, in 1986 and 1989, respectively, and the Ph.D. degree in philosophy from Universiti Putra Malaysia, in 2000. He is currently a Professor with the Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia. His research interests include artificial intelligence (pattern recognition in decision making with uncertainty–Bayesian reasoning; neural networks; fuzzy logic; and fuzzy neural networks, image processing–2D and 3D, edge detection, thinning, segmentation, feature extraction, image improvement, texture, resolution, image transforms, e.g., Trace, Fourier, and Wavelet) with applications to Jawi/Arabic manuscripts and biometric authentication.

**MOHAMMED ALSWATTI** (Member, IEEE) received the B.Eng. degree in computer engineering from the Islamic University of Gaza (IUG), Palestine, in 2010, the M.Sc. degree in electronic systems design engineering from Universiti Sains Malaysia (USM), Malaysia, in 2011, and the Ph.D. degree in computational intelligence under the Malaysian International Scholarship (MIS) Scheme. He worked as a Lecturer with the Software Engineering and Information Technology Faculties, University of Palestine, and an Instructor with the Educational Technology Centre, IUG. He also worked as a Graduate Assistant with the Electrical and Electronic Engineering Department, USM. He is currently working as an Assistant Professor with the School of Electrical and Computer Engineering, Xiamen University Malaysia (XMUM). His research interests include nature-inspired optimization-based clustering techniques, machine/deep learning applications, and pattern recognition.

**ISHAK HASHIM** was born in Melaka, Malaysia, in 1969. He received the B.Sc. degree in mathematics from The Ohio State University, Columbus, OH, USA, in June 1992, the M.Sc. degree in mathematics of nonlinear models from Heriot-Watt University, U.K., in November 1994, and the Ph.D. degree in mathematics from the University of Strathclyde, U.K., in March 1998, under the supervision of Prof. S. K. Wilson in fluid mechanics. He joined Universiti Kebangsaan Malaysia (UKM) as a Mathematics Tutor, in March 1993. He has been a Professor with the School of Mathematical Sciences, Faculty of Science and Technology, UKM, since August 2008. His research interests include applied mathematics include convexive instabilities in fluid flows, boundary layers, computational fluid dynamics, and analytical/numeric-analytic methods for nonlinear equations. **VOLUME 9, 2021**