Long-term memory Harris’ hawk optimization for high dimensional and optimal power flow problems

KASHIF HUSSAIN1, WILLIAM ZHU1, (SENIOR MEMBER, IEEE), AND MOHD NAJIB MOHD SALLEH2

1Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan, China.
2Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Batu Pahat, Johor, Malaysia.

Corresponding author: W. Zhu (e-mail: wfzhu@uestc.edu.cn).

This work was supported by the University of Electronic Science and Technology of China (UESTC) and National Natural Science Foundation of China (NSFC) under Grant No. 61772120.

ABSTRACT Harris’ hawk optimization (HHO) is a recent addition to population-based metaheuristic paradigm, inspired from hunting behavior of Harris’ hawks. It has demonstrated promising search behavior while employed on various optimization problems, however the diversity of search agents can be further enhanced. This paper represents a novel modified variant with a long-term memory concept, hence called long-term memory HHO (LMHHO), which provides information about multiple promising regions in problem landscape, for improvised search results. With this information, LMHHO maintains exploration up to a certain level even until search termination, thus produces better results than the original method. Moreover, the study proves that appropriate tools for in-depth performance analysis can help improve search efficiency of existing metaheuristic algorithms by making simple yet effective modification in search strategy. The diversity measurement and exploration-exploitation investigations prove that the proposed LMHHO maintains trade-off balance between exploration and exploitation. The proposed approach is investigated on high-dimensional numerical optimization problems, including classic benchmark and CEC’17 functions; also, on optimal power flow problem in power generation system. The experimental study suggests that LMHHO not only outperforms the original HHO but also various other established and recently introduced metaheuristic algorithms. Although, the research can be extended by implementing more efficient memory archive and retrieval approaches for enhanced results.

INDEX TERMS Diversity measurement, exploration-exploitation, long-term memory, Harris’ hawk optimization, optimal power flow

I. INTRODUCTION OPTIMIZATION is part of our routine problems, be it designing engineering structures, mining information from data science models, processing images and videos, finding optimal path in transportation, or achieving optimal flow of power in distributed systems – the case of this study. Usually, optimization is performed by choosing the best from a great deal of available solutions. It is achieved by finding best suitable parameters or decision variables that help reduce costs or maximize profits. However, optimization becomes significantly arduous when the size of decision variables surges exponentially; forming a high-dimensional optimization problem by expanding search-space immediately large. In these conditions, commonly used statistic methods often fail because of limited global searchability. Luckily, today, the field of optimization is inundated with innumerable global search optimization methods, called metaheuristic algorithms. Borrowing inspirations from almost every natural or man-made process, there is always a metaheuristic algorithm imitating one or the other metaphor. These algorithms have been often categorized into different groups [1]–[3], but the list seems unceasing. Sensing the deviation from true research direction, researchers have rightfully criticized the rampant inflow of new methods; insisted the need of more in-depth research, instead of building conclusions merely based on end-results [4]–[6].

Based on the discussion earlier, this research particularly calls for desperate measures in putting metaheuristic research
field in positive direction by utilizing and analyzing core search behaviors found in the metaheuristic algorithms that have already been introduced. It will be easily found that these methods often overlap one or the other search strategy [4]. There has been introduced plenty of algorithms devised with a range of search strategies, which if studied appropriately, can be further improved by effective modification or hybrid of two or more search strategies found in these algorithms and other deterministic methods [7]. Moreover, thorough analysis of the tripartite involving exploration, exploitation, and learning mechanisms with theoretical foundations and practical measurements will produce meaningful outcomes [8]. In this vein, this study performs in-depth performance analysis of one of the recently introduced promising metaheuristic algorithm Harris’ Hawk Optimization (HHO). The optimization method has already established its reputation with the help of applications in different areas [9]–[12]. The study performs population diversity measurement and exploitation-exploitation quantification for analyzing search behavior of the HHO algorithm. Based on extensive performance analysis, the research proposes long-term memory concept to be integrated in HHO so that a rigorous search can be performed in the problem landscape, especially when it is high-dimensional and non-convex. The proposed approach, namely long-term memory HHO (LMHHO) is evaluated on numerical optimization problems with different characteristics, as well as, on practical application of finding optimal power flow on IEEE bus system.

With increasing use of electrical gadgets and devices, the demand for energy production has risen tremendously. This requires continuous power generation in optimal conditions with the help of efficient operational planning for the thermal units. Optimal power flow (OPF) can be regarded as optimization problem where the operations of thermal power systems are optimized keeping in view certain physical and engineering constraints [13]. OPF problem is deemed crucial as it involves real-time adjustment of system parameters to meet energy demands, as well as, avoid possible breakdowns. It is a complex and difficult problem because it involves highly non-linear and non-convex functions. Moreover, the introduction of renewable energy in this paradigm is making it more complicated research direction [14], [15]. There has been put forward effort from researchers while solving different types of OPF problems using different approaches including gradient-based, statistical, heuristic, and metaheuristic methods [16], [17]. The use of gradient and statistical methods becomes impractical when employed on OPF problems on today’s power systems [18]. Therefore, because of efficient global searchability, in complex optimization landscapes, a variety of metaheuristic approaches have been successfully employed to solve OPF problems [19]. By finding optimal set of control variables, these techniques achieve objective functions by satisfying certain constraints associated with the systems. Some of the successful applications of metaheuristic algorithms in OPF domain include particle swarm optimization (PSO) [21], firefly algorithm (FA) [26], and whale optimization algorithm (WOA) [27], etc. However, to the best of authors’ knowledge, implementation of HHO in this research area is yet to be found in previous literature. Therefore, in this connection, this study can be considered as the first attempt to apply HHO on OPF problem. Moreover, to evaluate search efficiency of the proposed LMHHO in OPF domain, it is employed to optimize the objective functions related to power generation cost, emission, and power loss while simulated on IEEE-30 bus system.

To summarize, the contributions of this study are:

- LMHHO is proposed by integrating long-term memory concept in the original HHO algorithm for performing rigorous search in problem space. The archive of multiple promising regions of search space helps LMHHO maintain population diversity throughout search process. Hence, LMHHO is able to maintain balance between exploration and exploitation.
- LMHHO is tested on extensive test-bed consisting of ten classic benchmark functions and 29 complex functions of CEC’17 suite. The optimization problems in these experiments include low and high dimensional functions.
- LMHHO is implemented on optimal power flow (OPF) problem for IEEE-30 bus system. The OPF is solved by minimizing fuel cost, emission, and power loss.
- Compared with well established metaheuristic algorithms used in experiments of this study and from literature, it can be suggested that LMHHO performed efficiently on hard optimization problems. Moreover, the efficacy of long-term memory concept encourages to investigate its integration with various other metaheuristic algorithms in future studies.

This paper is organized as follows. The subsequent section makes a review of related work performed in the area of optimal power flow using metaheuristic techniques; followed by Sec. III which provides comprehensive detail on HHO, proposed modification in LMHHO, OPF problem formulation, and implementation of LMHHO on OPF problem. Sec. IV presents the experimental results which are discussed and analyzed in Sec. V, while the study is concluded in Sec. VI where potential future research directions are also highlighted.

II. RELATED WORK

To achieve minimum operational cost and maximum output, optimization is often required while operating power systems. Optimization problems, in this context, include economic power dispatch, combined heat and power dispatch, optimum scheduling of power generating units, optimal power flow in different systems like flexible alternating current transmission system (FACTS) devices, optimal AC-DC power flow, optimal reactive power flow, and load frequency control, etc. [20]. In literature, several optimization methods have been implemented to solve these problems. These include metaheuristic algorithms which show promis-
For solving OPF problems, different metaheuristic algorithms have been successfully employed. These include particle swarm optimization (PSO) [21]–[23], flower pollination algorithm (FPA) [24], moth-flame optimization (MFO) [25], firefly algorithm (FA) [26], whale optimization algorithm (WOA) [27], symbiotic organisms search (SOS) [28], jaya algorithm (JA) [29], grey wolf optimization (GWO) [30], backtracking search algorithm (BSA) [31], and many more. To the best of authors’ knowledge, HHO has not been implemented in any optimization problems of this category. A brief review of some important works, from recent literature, related to OPF and metaheuristic techniques is done as following.

In [20], Duman modified moth swarm optimization (MSA) method and applied it on solving optimal power flow problems in two-terminal HVDC systems with different objective functions. The Duman work achieved best fuel cost, voltage deviation, and voltage stability results as compared to counterparts used in this study. According to the researcher, the proposed arithmetic crossover approach in MSA produced enhanced search results and convergence to global optimum locations. The research contended to have achieved the trade-off balance between exploration and exploitation. In another application of MSA on OPF, Elattar [14] considered operational cost minimization, transmission power loss minimization, and improvement of voltage profile. Shilaja and Arunprasath [32] also considered MSA for solving OPF problems in IEEE-30, 57, and 118 bus systems with and without wind power resources. The research integrated MSA with gravitational search algorithm (GSA) for enhanced population diversity. Another moth inspired metaheuristic algorithm moth-flame optimization (MFO) [33] was improved and employed on achieving minimized results for objective functions considering fuel cost, gas emission, power loss, and voltage stability improvement. For evaluation, the proposed approach was simulated on three different test environments including IEEE-30, 57, and 118 bus systems.

In a recent study performed by Khunkitti et al. [18], a hybrid of dragon fly algorithm (DA) and PSO was employed on single and multi-objective OPF problems. The study used generation cost, emissions, and transmission loss as objective functions to be minimized while finding optimum decision variables for the standard IEEE-30 and 57 bus systems. The integration of exploration ability of DA and exploitation ability of PSO resulted in faster convergence and efficient results as compared to the relevant canonical methods and others from literature. Similarly, another attempt was made by Que and Wu [34] where a hybrid of bacterial foraging algorithm (BFA) and PSO was proposed for solving OPF problem using IEEE-30 bus system. The authors merely focused on fuel cost minimization, but achieved better results than the original BFA and PSO.

A novel approach to improved bat algorithm (BA) performance was proposed and applied on multi-objective OPF problem [35]. In this research, the authors improved local search ability of BA by using monotone random filling model based on extreme learning (MRFME), and they enhanced global search by mutation and crossover. The research also employed fuzzy based Pareto dominance method for achieving constrained Pareto optimal set. Total generation cost, emission, and power loss were used as evaluation criteria while simulating on IEEE-30, 57, and 118 bus systems. In [36], Duman solved OPF problem with and without valve point effect and prohibited zones; forming four different scenarios. The study used symbiotic organisms search (SOS) on power system with IEEE-30 bus. Results of the proposed SOS outperformed various other population-based and evolutionary algorithm from literature. The OPF problem with twelve case studies in wind and photovoltaic power generation systems were examined with single and multi-objective optimization. Based on simulations performed on IEEE-30 and 118 test systems, the research claimed to have achieved efficient results as compared to counterparts from literature. A better review of various metaheuristic techniques applied on OPF problems can be found in [20].

Apart from brief literature review presented earlier, the overall importance of this particular research area is briefly studied by applying keywords “optimal power flow” and (“optimal power flow” AND metaheuristic) on Scopus1 database which is widely used by research community. Fig. 1 shows that mostly OPF problems have been solved in engineering and energy followed by mathematics and computer science. Other research areas include business and social science, physics and material sciences, and environment science. While considering OPF research in timeline of last decade (until July 2019), Fig. 2 suggests that the interest from researchers from various backgrounds is increasing, as a constant rise can be observed in the number of publications. However, there is clearly significant gap for metaheuristic community to work in this particular research direction. In this short survey appeared three groups of approaches to solving OPF problems. These include machine learning methods, metaheuristic algorithms, and deterministic techniques (Fig. 3).

III. MATERIALS AND METHODS

This section elaborates on methodology adopted to investigate the proposed technique on high-dimensional optimization problems, as well as, OPF problem. The basic HHO algorithm is explained in the following subsection ahead of the proposed approach with long-term memory concept. Comes next the mathematical formulation of OPF problem, followed by implementation of the proposed method.

A. HARRIS’ HAWK OPTIMIZATION (HHO)

The HHO algorithm is a nature inspired population-based metaheuristic algorithm, based on the metaphor of prey capturing approach of the bird Harris’ hawk. Using “surprise pounce” tactic, the hawks attach prey from different

1https://www.scopus.com/search/form.uri?display=basic
FIGURE 1: OPF problems solved in different areas of research.

FIGURE 2: OPF research intensity in the last decade.

FIGURE 3: OPF methods used in literature.

FIGURE 4: HHO phases for performing search [9].

HHO purposes.

In metaheuristic algorithmic language, HHO launches search by initial random positions of the hawks which serve as candidate solution – representing a vector of decision variables to be optimized. Later, as the search proceeds, HHO turns from explorative to exploitative algorithm. Initially, HHO uses perching strategy to locate the prey on ground. Here, it is important to mention that the prey is a rabbit which is termed for the best location in search space found so far. The perching is modeled via Eq. (1). In Eq. (1), the first case represents scenario when hawks perch randomly within the space decided by the group, whereas the second case describes situation when the hawks perch around family members close to rabbit.

\[ x_{i}^{t+1} = \begin{cases} 
    x_{rand} - r_1|x_{rand} - 2r_2x_{avg}^t| & \text{if } q \geq 0.5 \\
    (x_{rabbit} - x_{avg}^t) - r_3[(ub_i + lb_i) + q(ub_i - lb_i)] & \text{if } q < 0.5 
\end{cases} \tag{1} \]

where \( x_i^t \) and \( x_i^{t+1} \) are respectively current position of the \( i \)th hawk and its new position in iteration \( t + 1 \), whereas \( x_{rand} \) and \( x_{rabbit} \) are respectively randomly selected hawk position and the best location (prey rabbit), and \( x_{avg}^t \) is dimension-wise average of \( N \) solution vectors. It is noteworthy that there are also several other ways to compute average vector in a matrix, including element-wise average. In Eq. (1), \( r_1 \) to \( r_4 \) and \( q \) are five different random numbers generated within the range \([0,1]\), whereas \( lb_i \) and \( ub_i \) are bounds of the search space.

The transition from exploration to exploitation phase is implemented with the idea of prey trying to escape the catch. The energy level of prey drops gradually during its escape...
attempt, this helps model convergence ability of HHO. Eq. (2) expresses mathematical modeling of the fact:

$$E = 2E_0 \left(1 - \frac{t}{t_{max}}\right)$$  \hspace{1cm} (2)$$

where $E_0$ and $E$ are initial and current energy levels of prey to escape, accordingly. In every iteration, the initial energy level $E_0$ alters randomly between [-1,1]. Interestingly in HHO, when $E_0$ decreases from 0 to -1, it exhibits that the rabbit’s energy is exhausting; and when $E_0$ increases from 0 to 1, it shows that the rabbit is gaining energy. Nevertheless, as the iterations progress, the current energy $E$ reduces. The HHO remains explorative as long as $|E| > 1$ and hawks keep on exploring global regions, whilst it turns into exploitative mode for exploiting on the already identified promising regions when $|E| < 1$.

The HHO algorithm ensures avoiding trapping in local optima or state of stagnancy by devising four different exploitation behaviors namely, soft besiege, hard besiege, soft besiege with progressive rapid dive, and hard besiege with rapid dive. In all these prey chasing styles, the hawks in HHO perform search around potential region in search space identified in exploration phase, using random number $r$ in range [0,1] and current energy level $E$. Soft besiege is when the prey rabbit is exhausted and has low energy to escape. Eq. (3) expresses mathematical modeling of the fact:

$$X_i^{t+1} = S \times \text{Lévy}(D) - E \left\lfloor J \cdot x_{rabbit} - x_i^t \right\rfloor$$

$$\Delta x_i^t = x_{rabbit} - x_i^t, \quad J = 2(1 - r_5)$$  \hspace{1cm} (3)$$

where $\Delta x_i^t$ is distance between the best location found so far and current position of $i$th hawk, and $r_5$ is a random number between [0,1] represents random jump of the rabbit trying to dodge the predator. When $|E| > 0.5$ and $r < 0.5$ then soft besiege is performed with progressive rapid dive (Fig. 5). It implies that the rabbit has enough energy to escape by making random zigzag moves and, in catch attempts, hawks make irregular rapid dives. In this situation, the hawks try progressive dives for best possible position to catch the prey. To model this, HHO utilizes Lévy flight approach. The next move of $i$th hawk, where it is making soft besiege with progressive dive, is formulated via Eq. (4):

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

$$Y = x_{rabbit} - E \left\lfloor J \cdot x_{rabbit} - x_i^t \right\rfloor,$$

$$Z = Y + S \times \text{Lévy}(D)$$  \hspace{1cm} (4)$$

where $D$ and $S$ are accordingly problem dimensions and random number vector of size $D$, whereas $f(Y)$ and $f(Z)$ are objective function values for the given vectors. The Lévy flight is formulated as Eq. (5):

$$\text{Lévy}(D) = 0.01 \times \frac{u \times \sigma}{|v|^{\beta}},$$

$$\sigma = \left( \frac{\Gamma(1+\beta) \times \sin \left( \frac{\pi \beta}{2} \right)}{\Gamma \left( \frac{1+\beta}{2} \right) \times 2^{\frac{\beta-1}{2}}} \right)$$  \hspace{1cm} (5)$$

where $u$ and $\beta$ are random numbers between [0,1] and a constant value (default $\beta = 1.5$), respectively. Notice another constant value of 0.01 in Eq. (5) used to control step length, which can be changed to adjust according to problem landscape.

Hard besiege is when $r \geq 0.5$ and $|E| < 0.5$, implies that the rabbit is exhausted and has low energy to escape. Eq. (6) models the situation:

$$X_i^{t+1} = x_{rabbit} - E |\Delta x_i^t|$$  \hspace{1cm} (6)$$

when $|E| < 0.5$ and $r < 0.5$ then the concept of hard besiege with progressive dive is implemented (Fig. 6). It means that the rabbit has significantly low energy that it cannot escape and hawks are close to make dive for successive catch. This concept is implemented using Eq. (7):
strategy highly selective. This single centrally guiding position for other population individuals may not guarantee convergence to global optimum location. The concept of single global best position often leads metaheuristic techniques suffer from premature convergence. That said, this research introduces long-term memory concept in HHO, where the population individuals can decide about the next move based on multiple past experiences. The idea provides broader view of multiple promising locations hence the chances of premature convergence or stagnancy are mitigated.

The proposed long-term memory HHO (LMHHO) annexes an extra parameter Memory Length (ML). ML is a control parameter defined by user that how many past experiences the swarm or population can remember at a time. However, it is important to mention that extra-large value for ML may result in storing unnecessarily large suboptimal information; therefore, the choice of this parameter should be made carefully according to problem landscape, or based on several trials. In this connection, sensitivity analysis related to ML parameter is performed in upcoming Sec. IV-A1. The process of updating long-term memory works as FIFO queue (first-in-first-out). In this memory, ML best locations found so far are stored. In FIFO, the queue is updated by appending the new item and removing the last item. In case of HHO, at every iteration \( t \), the memory is updated by appending the latest best location found so far and removing the oldest. When the memory is updated, the swarm makes next move based on any one item selected from long-term memory. The selection is made with the help of probability, and probability of selection \( p_i \) for \( i \)th item in memory is calculated as in Eq. (8):

\[
p_i = \frac{f(x^i_{rabbit})}{\sum_{j=1}^{ML} f(x^j_{rabbit})} \tag{8}
\]

where \( f(x^i_{rabbit}) \) or \( f(x^j_{rabbit}) \) is the fitness value of \( i \)th or \( j \)th item in long-term memory. Once probability of selection is calculated for each item in long-term memory, the selection is performed using Roulette Wheel Selection method. Now that an item from long-term memory is selected, it can be used in all position update equations. Meaning that, instead of single global best position \( x_{rabbit} \) used in HHO, LMHHO uses \( x^k_{rabbit} \) which is \( k \)th global best position in long-term memory. The index \( k \) is selected via Roulette Wheel Selection method. This way, all the position update equations in LMHHO are same as HHO except for replacing \( x_{rabbit} \) with \( x^k_{rabbit} \). The process of updating long-term memory is depicted via Fig. 8.

As per discussion earlier in this section, LMHHO brings simple yet effective modification in the original algorithm. This ascertains that appropriate in-depth analysis of metaheuristic algorithms may help produce efficient results; instead of introducing new algorithms every while, where dozens of others already exist. The MATLAB code of LMHHO has been made publicly available at

![FIGURE 7: LMHHO step by step schema.](image-url)
There are three fundamental components used to define computational complexity of any metaheuristic algorithm: population size $N$, problem dimensions $D$, and the number of iterations $T$. Moreover, the computational complexity of the algorithm is expressed with help of three basic processes: population initialization $O(N)$, objective function evaluation $O(N \times T)$, and population update $O(T \times N \times D)$. Hence, HHO computational complexity can be defined as Eq. (9):

$$O(HHO) = O(N) + O(N \times T) + O(T \times N \times D)$$

Note that LMHHO is modified with additional process of updating long-term memory $O(T \times ML \times D)$ and selection of items from the memory $O(T \times ML)$ for position update process, the algorithm complexity of LMHHO can be defined as Eq. (10):

$$O(LMHHO) = O(N) + O(N \times T) + O(N \times T) + O(T \times N \times D) + O(T \times ML \times D) + O(T \times ML)$$

1) Algorithm complexity

To better analyze search behavior of any optimization technique, it is essential to gauge exploration and exploitation, as the two contradicting characteristics influence performance. According to Hussain et al. [39], it is possible to measure explorative and exploitative capabilities once diversity in population is calculated. The research proposed dimension-wise diversity measurement as an effective technique which is employed in this current study. The population of $N$ candidate solutions with $D$ dimensional vectors can be depicted as Eq. (11):

$$X = (x_1, x_2, \ldots, x_N) = [x_{11}, x_{12}, \ldots, x_{1D}, x_{21}, x_{22}, \ldots, x_{2D}, \ldots, x_{N1}, x_{N2}, \ldots, x_{ND}]$$

where $X$ is population of candidate solutions with every $i$th solution vector $x_i \rightarrow \mathbb{R}$ is a set of real numbers of $D$ dimensional vector representing control variables of optimization problem in hand. When difference between dimensions of candidate solutions in the population increases, it means the algorithm is in exploration phase; otherwise, it is in exploitation phase if the difference narrows. Fig. 9 better illustrates the exploration and exploitation concept.

Mathematically, to measure diversity in population, Eq. (12) is used:

$$\bar{x}_j^t = \frac{\sum_{i=1}^{N} x_{ij}^t}{N}$$

$$Div_j^t = \frac{\sum_{i=1}^{N} \|\bar{x}_j^t - x_{ij}^t\|}{N}$$

$$Div^t = \frac{\sum_{j=1}^{D} Div_j^t}{D}$$

where $t \in \{1, 2, \ldots, t_{max}\}$

where $x_{ij}^t$ is $j$th dimension of $i$th candidate solution in population of $N$ in iteration $t$, $\bar{x}_j^t$ is average of dimension $j$, $Div_j^t$ represents diversity in $j$th dimension, and $Div^t$ is population diversity during iteration $t$ for total iterations $t_{max}$. Once population diversity measured in $Div^t$ for all the iterations, it is now feasible to measure exploration and exploitation percentage ratios during search process, using Eq. (13):

$$X_{pl}^t = \frac{Div^t}{\max(Div)} \times 100$$

$$X_{pt}^t = \frac{[Div^t - \max(Div)]}{\max(Div)} \times 100$$

where $X_{pl}^t$ and $X_{pt}^t$ are exploration and exploitation percentages, respectively, for iteration $t$, whereas $\max(Div)$ is the maximum population diversity in whole search process ($t_{max}$).

The MATLAB code for measuring population diversity and exploration-exploitation for population-based metaheuristic algorithms has been made publicly available at https://github.com/usitsoft/Exploration-Exploitation-Measurement.

C. OPTIMAL POWER FLOW PROBLEM (OPF)

Similar to other optimization problems, OPF also represents a problem where the best operating levels for power generation systems are achieved with the help of optimal set of parameters.
control parameters. It is a problem with complex non-linear and non-convex search space [15], [20]. Mathematically, an OPF is a minimization problem which can be formulated as Eq. (14):

\[
\begin{align*}
\text{min} & \ f(x, u), \text{ subject to} \\
& g_j(x, u) = 0, j \in [1, 2, \ldots, m] \\
& h_j(x, u) \leq 0, j \in [1, 2, \ldots, n]
\end{align*}
\]

(14)

where \( f(x, u) \) is objective function with \( x \) and \( u \) as state and control variables, respectively. The given objective function should be achieved by satisfying certain equality and inequality constraints. In Eq. (14), \( g_j(x, u) \) and \( h_j(x, u) \) are equality and inequality constrains representing, respectively, \( m \) power flow equations and \( n \) transmission and security limits. The state variable \( x \) is a vector defined as Eq. (15):

\[
x = [P_{G_{\text{slack}}}, V_{L1} \ldots V_{LNPQ}, Q_{G1} \ldots Q_{GN}, S_{I1} \ldots S_{NL}]
\]

(15)

where \( P_{G_{\text{slack}}} \) is generator active power at slack bus, \( V_L \) denotes voltage magnitude of load buses, \( Q_G \) represents generators reactive power, and \( S \) is the power of transmission lines. The vector \( u \) is defined as Eq. (16):

\[
u = [P_{G1}, \ldots P_{GN}, V_{G1} \ldots V_{GN}, T_{I1} \ldots T_{NT}, Q_{C1} \ldots Q_{CNC}]
\]

(16)

where \( P_G \) denotes active power output, \( V_G \) generator node voltage, \( T \) transformer tap ratio, and \( QC \) as active power injection.

1) Objective functions

To evaluate the proposed optimization method while solving OPF problem, three major objective functions are minimized in this study: basic fuel cost, emission, and power loss. The objective functions are defined as following:

a: Minimization of basic fuel cost

In this objective function, total fuel cost of the generators is calculated using Eq. (17):

\[
f_c = \left( \sum_{i=1}^{N_G} a_i P_{Gi}^2 + b_i P_{Gi} + c_i \right) + \text{Penalty}
\]

\[
\left( \frac{\$}{h} \right)
\]

(17)

where \( a_i, b_i, \) and \( c_i \) are defined as cost coefficients of the \( i \)th generator in total \( N_G \) generators. The fuel cost is calculated in the unit of dollar per hour.

b: Minimization of emission

In this objective functions, gases emission is minimized while generating power, using Eq. (18):

\[
f_e = \left( \sum_{i=1}^{N_G} a_i P_{Gi}^2 + b_i P_{Gi} + \gamma_i + \eta_i \exp(\delta P_{Gi}) \right) + \text{Penalty}
\]

\[
\left( \frac{\text{ton}}{h} \right)
\]

(18)

where \( a_i, b_i, \gamma_i, \) and \( \delta_i \) are defined as emission coefficients of the \( i \)th generation unit in total \( N_G \) thermal units.

c: Minimization of power loss

This objective function refers to minimization of power loss during power transmission in the network. Eq. (19) expresses the function mathematically:

\[
f_p = \left( \sum_{i=1}^{N_G} G_{i,i} \left[ V_i^2 + V_j^2 - 2V_i V_j \cos(\delta_i - \delta_j) \right] \right) + \text{Penalty(MW)}
\]

(19)

where \( G_{i,i} \) is conductance of \( i \)th transmission line connecting bus \( i \) and \( j \). \( V_i \) and \( V_j \) are voltage levels at bus \( i \) and \( j \), whereas \( \delta_i \) and \( \delta_j \) refer to voltage angles at bus \( i \) and \( j \).

2) Constraints

The objection functions mentioned earlier are subject to satisfy certain constraints which are explained as following. The Eq. (20) expresses equality constrains which are basically power flow equations:

\[
P_i - V_i \sum_{j=1}^{N_B} V_j (G_{ij} \cos(\delta_{ij}) + B_{ij} \sin(\delta_{ij})) = 0
\]

(20)

\[
Q_i - V_i \sum_{j=1}^{N_B} V_j (G_{ij} \sin(\delta_{ij}) - B_{ij} \cos(\delta_{ij})) = 0
\]

where \( N_B \) is the number of buses in power generation system. \( P_i \) and \( Q_i \) are respectively active power and reactive power at \( i \)th bus. \( \delta_{ij} \) is voltage angle between bus \( i \) and \( j \). \( G_{ij} \) and \( B_{ij} \) are respectively real and imaginary parts of bus admittance matrix of \( i \) by \( j \). The inequality constraints refer to limits of voltage level at different buses, power output, and branch flow. Eq. (21) shows these constraints:

\[
V_{Li}^{\text{min}} \leq V_{Li} \leq V_{Li}^{\text{max}}, i \in [1, 2, \ldots, N_L]
\]

\[
Q_{Gi}^{\text{min}} \leq Q_{Gi} \leq Q_{Gi}^{\text{max}}, i \in [1, 2, \ldots, N_G]
\]

\[
S_{li} \leq S_{li}^{\text{max}}, i \in [1, 2, \ldots, N_{TL}]
\]

(21)

The Eq. (22) defines space for the possible solutions for OPF problem:

\[
P_{Gi}^{\text{min}} \leq P_{Gi} \leq P_{Gi}^{\text{max}}, i \in [1, 2, \ldots, N]
\]

\[
V_{Li}^{\text{min}} \leq V_{Li} \leq V_{Li}^{\text{max}}, i \in [1, 2, \ldots, N_L]
\]

\[
T_{i}^{\text{min}} \leq T_i \leq T_{i}^{\text{max}}, i \in [1, 2, \ldots, N_T]
\]

\[
Q_{Gi}^{\text{min}} \leq Q_{Gi} \leq Q_{Gi}^{\text{max}}, i \in [1, 2, \ldots, N_G]
\]

(22)

3) Constraints handling

In this study, the penalized objective function is used to integrate the inequality variables so that the variables can be limited to avoid infeasible solutions. The penalty function is expressed as in Eq. (23):

\[
g(x, u) = f(x, u) + K_F (P_{G_{\text{slack}}} - P_{G_{\text{slack}}}^{\text{lim}})^2 +
\]

\[
K_V \sum_{i=1}^{N_L} (V_{Li} - V_{Li}^{\text{lim}})^2 + K_Q \sum_{i=1}^{N_G} (Q_{Gi} - Q_{Gi}^{\text{lim}})^2 +
\]

\[
K_S \sum_{i=1}^{N_T} (S_{li} - S_{li}^{\text{lim}})^2
\]

(23)

where \( K_F, K_V, K_Q, \) and \( K_S \) are penalty factors.
D. OPF IMPLEMENTATION USING LMHHO

The steps of application of LMHHO on OPF problem are listed in Algorithm 1:

Algorithm 1 OPF implementation using LMHHO

Initialization:
- Load system data; such as, coefficients for fuel cost and emission, initial values of active power of generators, node voltages, transformer tap ratio, upper and lower limits of constraints (\(S_{\text{li}}, P_{G_{i}}, V_{G_{i}}, V_{L_{i}}, Q_{G_{i}}, \text{and } T_{i}\))
- LMHHO control parameters (population size \(N\), long-term memory limit \(ML\), initial rabbit energy \(E_{0}\), maximum iterations \(t_{\text{max}}\))
- Initialize hawks population with random positions in search space

Convert multiobjective OPF into unconstrained problem via Eq. (23)

Evaluate fitness of initial population via selected objective function

Find \(ML\) best positions and assign long-term memory \([x_{\text{rabbit}}^{1}, x_{\text{rabbit}}^{2}, \ldots, x_{\text{rabbit}}^{ML}]\)

while \(t \leq t_{\text{max}}\) do

Update escaping energy of each rabbit via Eq. (2)

Select item from long-term memory:

Calculate probability of selection using Eq. (8)

Use Roulette Wheel Selection method to select an item from long-term memory

for each hawk \(x_{i}\) do

if \(|E| \geq 1\) then

Update position \(x_{i}^{t+1}\) using Eq. (1)

else

if \(i \leq 3\) then

Update position \(x_{i}^{t+1}\) using Eq. (3)

end if

if \(r \geq 0.5\) and \(|E| < 0.5\) then

Update position \(x_{i}^{t+1}\) using Eq. (6)

end if

if \(r < 0.5\) and \(|E| \geq 0.5\) then

Update position \(x_{i}^{t+1}\) using Eq. (4)

end if

if \(r < 0.5\) and \(|E| < 0.5\) then

Update position \(x_{i}^{t+1}\) using Eq. (7)

end if

end if

Evaluate population fitness via selected objective function \{\(f_{c}\) Eq. (17), \(f_{c}\) Eq. (18), or \(f_{p}\) Eq. (19)\}

end for

Update iteration counter \(t = t + 1\)

end while

return best solution from long-term memory

\[\min[x_{\text{rabbit}}^{1}, x_{\text{rabbit}}^{2}, \ldots, x_{\text{rabbit}}^{ML}]\]


IV. EXPERIMENTAL STUDY

To investigate performance of the proposed LMHHO with long-term memory concept, this study employed two types of problems: benchmark numerical optimization problems and real-world application of solving OPF problem. The detail of each experiment is given in the following subsections. All the experiments were performed in the programming tool MATLAB 2017b on computing environment with Intel® Core™ i5 (3.40 GHz) CPU with RAM 8 GB, and operating system Microsoft Windows 10. Because metaheuristic algorithms are stochastic in nature and running them multiple times may generate varying results, therefore an average of multiple runs is considered for performance evaluation. In this study, the metaheuristic algorithms used in this study were run 30 times on benchmark problems and 10 times on OPF problems. For each of the two experimental cases, Table 1 presents parameter settings for the used metaheuristic algorithms.

A. EXPERIMENTAL STUDY 1: NUMERICAL OPTIMIZATION PROBLEMS

1) Classic benchmark functions

The efficiency of the proposed LMHHO was evaluated on ten commonly used benchmark functions that represent unimodal and multimodal optimization problems. The unimodal functions are used to evaluate convergence ability as these problems maintain several single global best location and no local optimal regions. On the other hand, multimodal test functions maintain several widespread local minima in the landscape, with single global best location, these methods are used to examine global searchability of the optimization method. It is therefore, these two types of optimization problems benchmark exploration and exploitation abilities of the search strategies in a metaheuristic algorithm. Moreover, most of these problems often represent real-life optimization problems [38]. Considering the low as well as high dimensional problems, this study employed the selected numerical problems with \(D = 30\) and \(D = 500\) dimensions, to prove the robustness of the proposed approach on highly non-convex optimization hypercubes. Table 2 lists the employed 
numerical optimization problems.

### TABLE 2: Benchmark test functions (U=Unimodal, M=Multimodal, S=Separable, N=Non-separable, C=Continuous)

<table>
<thead>
<tr>
<th>Function (USS)</th>
<th>Range</th>
<th>( f(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sphere (S)</td>
<td>([0, 100]^D)</td>
<td>0</td>
</tr>
<tr>
<td>Powell Singular 1 (S)</td>
<td>([-4, 5]^D)</td>
<td>0</td>
</tr>
<tr>
<td>Schwefel 2.22 (S)</td>
<td>([-10, 10]^D)</td>
<td>0</td>
</tr>
<tr>
<td>Sum-Squares (USS)</td>
<td>([-500, 500]^D)</td>
<td>0</td>
</tr>
<tr>
<td>Ackley (MC)</td>
<td>([-500, 600]^D)</td>
<td>0</td>
</tr>
<tr>
<td>Griewank (MC)</td>
<td>([-1000, 1000]^D)</td>
<td>0</td>
</tr>
<tr>
<td>Rastrigin (MCN)</td>
<td>([0, 10]^D)</td>
<td>0</td>
</tr>
<tr>
<td>Xin-Shu Yang-2 (MCN)</td>
<td>([-2\pi, \pi]^D)</td>
<td>0</td>
</tr>
<tr>
<td>Zakharov (MCN)</td>
<td>([-5, 10]^D)</td>
<td>0</td>
</tr>
</tbody>
</table>

Since LMHHO uses \( ML \) to determine the length of long-term memory in the algorithm, hence selection of this parameter is crucial to its performance. The extra-ordinary memory length will store higher number of best solutions found so-far, which may unnecessary divert the search agents on undesirable regions – resulting ineffective search performance. On the other hand, shorter memory length will store less information about past experience, which may lead to underperforming scenario. It is, therefore, important to determine the value of \( ML \) parameter sensibly, as it is sensitive to LMHHO efficiency. The sensitivity analysis of \( ML \) parameter was performed in this study. The results are given in Table 3. The analysis was performed on 500-dimensional functions listed in Table 2, for parameter value 5, 10, and 15. According to statistics in Table 3, it can be suggested that the best suitable value for \( ML \) parameter can be set to 10. The search efficiency of LMHHO was ineffective with \( ML = 5 \), whereas it was similar with \( ML = 10 \) and \( ML = 15 \). Therefore, it can be inferred that \( ML = 10 \) seems to be suitable value for LMHHO algorithm. Despite sensitivity analysis performed in this study, the value of \( ML \) parameter is still problem dependent, thus should be chosen based on problem complexity.

For effective in-depth performance analysis, this study measured diversity and exploitation-exploitation ratios while solving these problems. Moreover, statistical results are also presented for robust conclusions. For comparisons, apart from LMHHO and HHO, other established and recently introduced metaheuristic algorithms were also used in experiments. The competitive algorithms include PSO [40], bernstein-search differential evolution (BSDE) [54], artificial bee colony (ABC) [41], firefly algorithm (FA) [42], elephant herding optimization (EHO) [46], thermal exchange optimization (TEO) [47], grasshopper optimization algorithm (GOA) [48], and grey wolf optimization (GWO) [49].

Table 4 provides diversity and exploration-exploitation measurement in HHO and LMHHO while solving numerical optimization problems with \( D = 30 \) and \( D = 500 \). A graphical presentation of the population diversity and exploration-exploitation phases is given in Figs. 10, 11, 12, 13, 14 and 15. Moreover, statistical results achieved by the selected algorithms are presented in Tables 5 and 6 for \( D = 30 \) and \( D = 500 \) problems, respectively. On the other hand, convergence abilities of the investigated methods are depicted in Fig. 16 for \( D = 30 \) and in Fig. 17 for \( D = 500 \) problems.

According to Table 4, the proposed modification in LMHHO mostly reduced population diversity in the algorithm. For example, on \( f_1 \) LMHHO maintained population diversity value 45.26 \((D = 30)\) and 755.23 \((D = 500)\) which is relatively lesser than diversity values 45.88 \((D = 30)\) and 790.90 \((D = 500)\) in HHO. Similarly, diversity measurement in LMHHO for functions \( f_2, f_3, f_4, f_5, f_6, \) and \( f_{10} \) with \( D = 30 \) remained lower than original HHO. However,
LMHHO maintained diversity slightly higher than HHO on $f_6$, $f_7$, and $f_8$ with $D = 30$. In these functions, diversity measurement for LMHHO was recorded as 45.72%, 46.89%, and 46.05 compared to 42.91%, 43.89%, and 44.29 diversity recorded for HHO. However, for functions $f_4$, $f_6$, $f_7$, and $f_9$ with $D = 500$, increase in population diversity was recorded for LMHHO. Moreover, Table 4 also reveals that mostly LMHHO maintained exploration percentage lower than exploitation on most of the functions with $D = 30$ and $D = 500$. For instance, LMHHO maintained exploration percentage as 27%, 29%, 28%, 22%, 21%, and 32% for functions $f_1$, $f_2$, $f_3$, $f_5$, $f_6$, and $f_{10}$ with $D = 30$; and these values are less than exploration measurements recorded for HHO. Similarly, on $D = 500$, LMHHO maintained exploration less than HHO. From Table 4, it can be observed that mostly LMHHO maintained exploration around 20% – 30% and exploitation around 70% – 80%. Likewise, in most of the functions with $D = 30$ and $D = 500$, HHO also maintained exploration lower than exploitation percentage, but it kept exploration percentage around 35% – 45% and exploitation percentage around 55% – 65%. This discussion can be further comprehended via Figs. 10 and 11 for diversity measurement and Figs. 12, 13, 14 and 15 for exploration and exploitation behaviors in original and the proposed modified variant.

According to statistical results of the selected algorithms on numerical problems with $D = 30$ and $D = 500$ provided in Tables 5 and 6, the proposed LMHHO performed efficiently on benchmark suite of $f_1$ to $f_{10}$. The results reveal that LMHHO mostly achieved global optimum values on 8 functions ($f_1$, $f_2$, $f_3$, $f_5$, $f_6$, $f_7$, $f_8$ and $f_{10}$) with $D = 30$ and $D = 500$, considering $8.88E-16$ as zero in case of Ackley function ($f_6$). Comparing results of LMHHO and HHO, it can be inferred that the proposed approach achieved far better results than the original method. For instance, LMHHO achieved $0.00E+00$ as compared to $3.61E-100$ generated by HHO for Sphere function $f_1$ with $D = 30$. 

![FIGURE 10: Diversity measurement of HHO and LMHHO on benchmark problems ($D = 30$)](image1)

![FIGURE 11: Diversity measurement of HHO and LMHHO on benchmark problems ($D = 500$)](image2)
On Schewefel’s 2.22 \((f_3)\) with \(D = 30\), LMHHO produced significantly better optimum value \(9.43E - 242\) compared to \(1.08E - 52\) by HHO. In case of \(D = 500\) functions also, LMHHO produced markedly promising results than original HHO algorithm. As in case of Sphere \((f_1)\) and Zakhvov \((f_{10})\) with \(D = 500\), LMHHO achieved global optimum values of \(0.00E + 00\) compared to HHO values \(8.67E - 94\) and \(1.08E + 03\) respectively for \(f_1\) and \(f_{10}\) functions. Besides LMHHO, other algorithms which achieved relatively better results were TEO and HHO. However, the least performers in these experiments were PSO, ABC, and EHO.

For statistical validation of the results discussed earlier, Wilcoxon rank sum test was carried out on these results. The related \(p\)-values are reported in Tables 5 and 6 along with symbols “+”, “=”, “-” which indicate significance of difference between the results of LMHHO and the competitors’, with 5% significance level. The \(p\)-value less than 0.05 suggests that the performance of LMHHO is significantly different from the method being compared, whereas the superiority of LMHHO over others is indicated by “+”, inferiority by “-”, while performance with insignificant difference is denoted by “=” symbol. The last row of the Tables 5 and 6 counts the number of cases LMHHO beats, losses or equals the competitor in terms of objective function values achieved. According to these counts in Table 5 for 30 dimensional functions, LMHHO results were significantly better than PSO and ABC on 9 out of 10 functions, whereas it equalized these methods on \(f_5\) function. LMHHO stood winner on all 10 functions against BSDE, FA, GOA, and HHO. When compared with TEO and HHO, LMHHO won on 7 and 6 functions, respectively, though it equalized on 3 and 4 functions accordingly.

The statistical test results related to 500 dimensional functions also suggest the superiority of LMHHO performance over others. LMHHO remained significantly better algorithm than PSO and ABC on 9 out of 10 functions, except for \(f_5\).
where all three performed equally well. On the other hand, when compared with BSDE, FA, EHO, GOA, and EHO, the related p-values are less than 0.05 suggesting LMHHO as winner on all 10 functions. There was only one method TEO which outperformed LMHHO on only one function $f_3$, and remained equally better on $f_6$ to $f_8$, and performed inferior to LMHHO on remaining 6 functions. HHO also underperformed than LMHHO on 7 functions and produced equally well results than LMHHO on 3 functions. From the overall impression drawn from Wilcoxon rank sum test on results related 30 and 500-dimensional functions, it can be suggested that LMHHO mostly remained significantly better than the counterparts on this test suit.

The computational cost is measured for LMHHO and the competitors on the functions listed in Table 2 with 500 dimensions. The measurement in seconds is given in Table 8. According to Table 8, the cost of LMHHO remained relatively higher than original algorithm because of additional processing of memory related tasks, as discussed in Sec. III-B1. However, when compared with other algorithms, there cannot be seen significant difference in computational cost of LMHHO and others. In fact, LMHHO remained relatively cheaper than FA, EHO, and GOA on most of the functions, yet it produced remarkably better results than these algorithms.

When analyzing convergence ability of the proposed LMHHO, Figs. 16 and 17 show that it found far better optimum locations as compared to standard HHO and other metaheuristic algorithms selected in this study. Moreover, LMHHO converged significantly faster compared to competitive approaches.

2) CEC’17 functions

For further validating efficiency of the proposed LMHHO on complex functions, CEC 2017 [53] problems with 100 dimensions were used. The CEC’17 test suite comprises

![Figure 14: Exploration-exploitation measurement of LMHHO on benchmark problems ($D = 30$)](image1)

![Figure 15: Exploration-exploitation measurement of LMHHO on benchmark problems ($D = 500$)](image2)
FIGURE 16: Convergence of the selected algorithms on benchmark problems ($D = 30$)

FIGURE 17: Convergence of the selected algorithms on benchmark problems ($D = 500$)

of complex benchmark functions which can be categorized into four types: $f_1$ to $f_3$ are unimodal, $f_4$ to $f_{10}$ are multimodal, $f_{11}$ to $f_{20}$ are hybrid functions, whereas $f_{21}$ to $f_{30}$ are composition functions. The related experimental result are presented in Table 7, in terms of mean and standard deviation of objective function values achieved over 30 independent runs. Moreover, the two-sided Wilcoxon rank sum test with 95% significance level also verifies the significance of outcome generated by LMHHO when compared with every other method used in experiments. A p-value less than 0.05 accepts alternative hypothesis and suggests that there is significant difference between the two methods. To indicate LMHHO as superior than the other competitive algorithm, "+" symbol is used, whereas "-" is used to refer the competitive algorithm as superior than LMHHO. The symbol "=" shows that LMHHO generated approximately similar outcome or both the approaches generated exactly same output.

According to results of CEC'17, LMHHO outperformed rest of the algorithms on most of the functions. Generally, the proposed approach achieved better results than others on 20 out of 29 functions. LMHHO outperformed PSO, BSDE, and TEO on all CEC'17 functions. While, it generated inferior results than ABC on one function ($f_6^{CEC}$), FA on three functions ($f_6^{CEC}$, $f_{16}^{CEC}$ and $f_{25}^{CEC}$), GOA on two functions ($f_{22}^{CEC}$ and $f_{25}^{CEC}$), GWO on one function ($f_1^{CEC}$), and HHO on two functions ($f_1^{CEC}$ and $f_{27}^{CEC}$). On the other hand, the proposed approach did not perform significantly different than ABC on two functions ($f_6^{CEC}$ and $f_{16}^{CEC}$), FA on two functions ($f_6^{CEC}$ and $f_{25}^{CEC}$), GOA on one function ($f_2^{CEC}$), and HHO on three functions ($f_{16}^{CEC}$, $f_{18}^{CEC}$, and $f_{27}^{CEC}$). Overall, it can be suggested from these experiments that the concept of long-term memory in LMHHO enhanced searchability of the algorithm, and managed to maintain exploration and exploitation balance. Though, it needs more research on further improvement in memory archiving and retrieving strategies.
### TABLE 5: Numerical optimization results on benchmark functions (D = 30)

<table>
<thead>
<tr>
<th>Func. Stats</th>
<th>PSO</th>
<th>GOA</th>
<th>HWO</th>
<th>9.43E-242</th>
</tr>
</thead>
<tbody>
<tr>
<td>f1 Mean</td>
<td>1.02E+03</td>
<td>2.95E+03</td>
<td>3.05E+03</td>
<td>1.19E-07</td>
</tr>
<tr>
<td>Std</td>
<td>3.16E+02</td>
<td>8.25E+03</td>
<td>9.01E+04</td>
<td>1.60E-08</td>
</tr>
<tr>
<td>p Val.</td>
<td>1.17E-12</td>
<td>5.13E-12</td>
<td>9.38E-13</td>
<td>1.17E-12</td>
</tr>
<tr>
<td>f2 Mean</td>
<td>1.83E+03</td>
<td>5.00E+03</td>
<td>4.26E+03</td>
<td>9.20E-09</td>
</tr>
<tr>
<td>Std</td>
<td>1.65E+03</td>
<td>5.00E+03</td>
<td>4.26E+03</td>
<td>9.20E-09</td>
</tr>
<tr>
<td>p Val.</td>
<td>1.17E-12</td>
<td>5.13E-12</td>
<td>9.38E-13</td>
<td>1.17E-12</td>
</tr>
<tr>
<td>f3 Mean</td>
<td>5.08E+01</td>
<td>2.55E+01</td>
<td>3.90E+01</td>
<td>4.52E+01</td>
</tr>
<tr>
<td>Std</td>
<td>2.32E+01</td>
<td>4.33E+00</td>
<td>9.46E+00</td>
<td>1.05E-05</td>
</tr>
<tr>
<td>p Val.</td>
<td>2.87E-11</td>
<td>2.68E-11</td>
<td>5.82E-12</td>
<td>8.12E-04</td>
</tr>
<tr>
<td>f4 Mean</td>
<td>1.16E+03</td>
<td>4.38E+02</td>
<td>1.68E+02</td>
<td>2.55E-01</td>
</tr>
<tr>
<td>Std</td>
<td>3.34E+02</td>
<td>1.22E+02</td>
<td>4.00E+01</td>
<td>1.05E-04</td>
</tr>
<tr>
<td>p Val.</td>
<td>1.17E-12</td>
<td>5.13E-12</td>
<td>9.38E-13</td>
<td>1.17E-12</td>
</tr>
<tr>
<td>f5 Mean</td>
<td>5.30E+01</td>
<td>4.52E+01</td>
<td>5.30E+01</td>
<td>5.21E-01</td>
</tr>
<tr>
<td>Std</td>
<td>1.22E+01</td>
<td>4.33E+01</td>
<td>9.46E+00</td>
<td>1.05E-05</td>
</tr>
<tr>
<td>p Val.</td>
<td>2.87E-11</td>
<td>2.68E-11</td>
<td>5.82E-12</td>
<td>8.12E-04</td>
</tr>
<tr>
<td>f6 Mean</td>
<td>1.83E+03</td>
<td>9.35E+03</td>
<td>3.04E+03</td>
<td>5.21E-01</td>
</tr>
<tr>
<td>Std</td>
<td>3.01E-11</td>
<td>5.13E-12</td>
<td>9.38E-13</td>
<td>1.17E-12</td>
</tr>
<tr>
<td>p Val.</td>
<td>3.01E-11</td>
<td>5.13E-12</td>
<td>9.38E-13</td>
<td>1.17E-12</td>
</tr>
<tr>
<td>f7 Mean</td>
<td>1.17E+00</td>
<td>2.78E+02</td>
<td>9.64E+02</td>
<td>3.45E-03</td>
</tr>
<tr>
<td>Std</td>
<td>3.01E-11</td>
<td>5.13E-12</td>
<td>9.38E-13</td>
<td>1.17E-12</td>
</tr>
<tr>
<td>p Val.</td>
<td>3.01E-11</td>
<td>5.13E-12</td>
<td>9.38E-13</td>
<td>1.17E-12</td>
</tr>
<tr>
<td>f8 Mean</td>
<td>1.83E+03</td>
<td>9.35E+03</td>
<td>3.04E+03</td>
<td>5.21E-01</td>
</tr>
<tr>
<td>Std</td>
<td>3.01E-11</td>
<td>5.13E-12</td>
<td>9.38E-13</td>
<td>1.17E-12</td>
</tr>
<tr>
<td>p Val.</td>
<td>3.01E-11</td>
<td>5.13E-12</td>
<td>9.38E-13</td>
<td>1.17E-12</td>
</tr>
<tr>
<td>f9 Mean</td>
<td>1.17E+00</td>
<td>2.78E+02</td>
<td>9.64E+02</td>
<td>3.45E-03</td>
</tr>
<tr>
<td>Std</td>
<td>3.01E-11</td>
<td>5.13E-12</td>
<td>9.38E-13</td>
<td>1.17E-12</td>
</tr>
<tr>
<td>p Val.</td>
<td>3.01E-11</td>
<td>5.13E-12</td>
<td>9.38E-13</td>
<td>1.17E-12</td>
</tr>
<tr>
<td>f10 Mean</td>
<td>1.17E+00</td>
<td>2.78E+02</td>
<td>9.64E+02</td>
<td>3.45E-03</td>
</tr>
<tr>
<td>Std</td>
<td>3.01E-11</td>
<td>5.13E-12</td>
<td>9.38E-13</td>
<td>1.17E-12</td>
</tr>
<tr>
<td>p Val.</td>
<td>3.01E-11</td>
<td>5.13E-12</td>
<td>9.38E-13</td>
<td>1.17E-12</td>
</tr>
</tbody>
</table>

### TABLE 6: Numerical optimization results on benchmark functions (D = 500)

<table>
<thead>
<tr>
<th>Func. Stats</th>
<th>PSO</th>
<th>GOA</th>
<th>HWO</th>
<th>9.88E-16</th>
</tr>
</thead>
<tbody>
<tr>
<td>f1 Mean</td>
<td>1.36E+04</td>
<td>5.05E+04</td>
<td>1.52E+04</td>
<td>4.92E+04</td>
</tr>
<tr>
<td>Std</td>
<td>1.13E+04</td>
<td>4.01E+04</td>
<td>3.51E+04</td>
<td>1.21E+04</td>
</tr>
<tr>
<td>p Val.</td>
<td>1.12E+02</td>
<td>9.38E-13</td>
<td>1.78E-12</td>
<td>1.69E+02</td>
</tr>
<tr>
<td>f2 Mean</td>
<td>5.30E+05</td>
<td>6.53E+05</td>
<td>7.45E+05</td>
<td>9.35E+03</td>
</tr>
<tr>
<td>Std</td>
<td>3.96E+04</td>
<td>3.50E-03</td>
<td>3.24E-03</td>
<td>1.94E+09</td>
</tr>
<tr>
<td>p Val.</td>
<td>1.17E+02</td>
<td>9.38E-13</td>
<td>1.78E-12</td>
<td>1.69E+02</td>
</tr>
<tr>
<td>f3 Mean</td>
<td>5.30E+04</td>
<td>8.59E+04</td>
<td>1.05E+04</td>
<td>1.29E+04</td>
</tr>
<tr>
<td>Std</td>
<td>3.96E+04</td>
<td>3.50E-03</td>
<td>3.24E-03</td>
<td>1.94E+09</td>
</tr>
<tr>
<td>p Val.</td>
<td>1.17E+02</td>
<td>9.38E-13</td>
<td>1.78E-12</td>
<td>1.69E+02</td>
</tr>
<tr>
<td>f4 Mean</td>
<td>5.30E+04</td>
<td>8.59E+04</td>
<td>1.05E+04</td>
<td>1.29E+04</td>
</tr>
<tr>
<td>Std</td>
<td>3.96E+04</td>
<td>3.50E-03</td>
<td>3.24E-03</td>
<td>1.94E+09</td>
</tr>
<tr>
<td>p Val.</td>
<td>1.17E+02</td>
<td>9.38E-13</td>
<td>1.78E-12</td>
<td>1.69E+02</td>
</tr>
<tr>
<td>f5 Mean</td>
<td>5.30E+04</td>
<td>8.59E+04</td>
<td>1.05E+04</td>
<td>1.29E+04</td>
</tr>
<tr>
<td>Std</td>
<td>3.96E+04</td>
<td>3.50E-03</td>
<td>3.24E-03</td>
<td>1.94E+09</td>
</tr>
<tr>
<td>p Val.</td>
<td>1.17E+02</td>
<td>9.38E-13</td>
<td>1.78E-12</td>
<td>1.69E+02</td>
</tr>
<tr>
<td>f6 Mean</td>
<td>5.30E+04</td>
<td>8.59E+04</td>
<td>1.05E+04</td>
<td>1.29E+04</td>
</tr>
<tr>
<td>Std</td>
<td>3.96E+04</td>
<td>3.50E-03</td>
<td>3.24E-03</td>
<td>1.94E+09</td>
</tr>
<tr>
<td>p Val.</td>
<td>1.17E+02</td>
<td>9.38E-13</td>
<td>1.78E-12</td>
<td>1.69E+02</td>
</tr>
<tr>
<td>f7 Mean</td>
<td>5.30E+04</td>
<td>8.59E+04</td>
<td>1.05E+04</td>
<td>1.29E+04</td>
</tr>
<tr>
<td>Std</td>
<td>3.96E+04</td>
<td>3.50E-03</td>
<td>3.24E-03</td>
<td>1.94E+09</td>
</tr>
<tr>
<td>p Val.</td>
<td>1.17E+02</td>
<td>9.38E-13</td>
<td>1.78E-12</td>
<td>1.69E+02</td>
</tr>
<tr>
<td>f8 Mean</td>
<td>5.30E+04</td>
<td>8.59E+04</td>
<td>1.05E+04</td>
<td>1.29E+04</td>
</tr>
<tr>
<td>Std</td>
<td>3.96E+04</td>
<td>3.50E-03</td>
<td>3.24E-03</td>
<td>1.94E+09</td>
</tr>
<tr>
<td>p Val.</td>
<td>1.17E+02</td>
<td>9.38E-13</td>
<td>1.78E-12</td>
<td>1.69E+02</td>
</tr>
<tr>
<td>f9 Mean</td>
<td>5.30E+04</td>
<td>8.59E+04</td>
<td>1.05E+04</td>
<td>1.29E+04</td>
</tr>
<tr>
<td>Std</td>
<td>3.96E+04</td>
<td>3.50E-03</td>
<td>3.24E-03</td>
<td>1.94E+09</td>
</tr>
<tr>
<td>p Val.</td>
<td>1.17E+02</td>
<td>9.38E-13</td>
<td>1.78E-12</td>
<td>1.69E+02</td>
</tr>
<tr>
<td>f10 Mean</td>
<td>5.30E+04</td>
<td>8.59E+04</td>
<td>1.05E+04</td>
<td>1.29E+04</td>
</tr>
<tr>
<td>Std</td>
<td>3.96E+04</td>
<td>3.50E-03</td>
<td>3.24E-03</td>
<td>1.94E+09</td>
</tr>
<tr>
<td>p Val.</td>
<td>1.17E+02</td>
<td>9.38E-13</td>
<td>1.78E-12</td>
<td>1.69E+02</td>
</tr>
</tbody>
</table>

### B. EXPERIMENTAL STUDY 2: POWER FLOW OPTIMIZATION PROBLEM

The real-life application for evaluating LMHHO was implementation on OPP problem. Here, simulations were performed...
TABLE 7: Numerical optimization results on CEC’17 functions (D = 10)

| Func. | CECC | BCEC | ARD | FA | EHO | GTOA | GO | BBO | LHAD
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>2.69E-11</td>
<td>2.60E+00</td>
<td>1.08E+02</td>
<td>8.97E+03</td>
<td>1.01E+04</td>
<td>2.82E-11</td>
<td>8.03E+02</td>
<td>5.10E+03</td>
<td>2.64E+11</td>
</tr>
<tr>
<td>Std.</td>
<td>2.68E-11</td>
<td>2.60E+00</td>
<td>1.08E+02</td>
<td>8.97E+03</td>
<td>1.01E+04</td>
<td>2.82E-11</td>
<td>8.03E+02</td>
<td>5.10E+03</td>
<td>2.64E+11</td>
</tr>
<tr>
<td>p-Val.</td>
<td>3.02E-11</td>
<td>3.02E-11</td>
<td>3.02E-11</td>
<td>3.02E-11</td>
<td>3.02E-11</td>
<td>3.02E-11</td>
<td>3.02E-11</td>
<td>3.02E-11</td>
<td>3.02E-11</td>
</tr>
</tbody>
</table>

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2946664, IEEE Access
Table 10. The best values for control variables achieved by the proposed LMHHO and other competitive algorithms are presented in Table 11, comparison of end results are made in Table 12. The competitive algorithms used in Table 11 are MGOA [19] and IMFO [33], whereas Table 12 contains competitive methods including MGOA [19], GOA [48], genetic algorithms (GA) [19], PSO [19], teaching-learning-based optimization (TLBO) [19], improved moth-flame optimization (IMFO) [33]. The convergence abilities of the LMHHO and HHO on OPF problems are illustrated via Fig. 19.

V. DISCUSSION AND ANALYSIS

According to the results provided in previous section, it can be affirmed that the proposed modification in LMHHO helped achieve significantly better global optimum output. On both the benchmark numerical problems as well as OPF problems, LMHHO showed considerable improvement. LMHHO not only outperformed existing HHO algorithm but also various other metaheuristic techniques which are inclusive of popular and recently introduced approaches. Tables 5, 6, and 7 suggest that LMHHO produced remarkably efficient results. This is consistent on OPF problems as well, where LMHHO minimized constrained optimization objectives better than original method, as well as, other those selected from recent literature; like MGOA, TLBO, and IMFO (Table 12). Fig. 19 depicts convergence characteristics of original HHO and LMHHO methods, which shows that the proposed modification improved convergence ability.

3) Case 3: Power loss minimization results

In this particular case, the proposed LMHHO minimized power loss through transmission lines, using objective function defined in Eq. (19). LMHHO achieved power loss value 3.00853 MW which is better than original HHO and IMFO according to Table 11, however MGOA achieved slightly better value than LMHHO. On the other hand, in comparison with GA, PSO, TLBO, and IMFO which generated values in MW respectively as 3.3141, 3.1342, 3.1079, 3.1202, and 3.0905, LMHHO achieved the lowest power loss value (Table 12). Fig. 19 depicts convergence characteristics of original HHO and LMHHO methods, which shows that the proposed modification improved convergence ability.
with ample options of optimal locations where further search could be made.

This discussion can be further validated from exploration-exploitation graphs presented in Figs. 12, 13, 14 and 15 for both low and high-dimensional problems. Generally, from the graphs, it is obvious that HHO started with high exploration and low exploitation; and later after middle of the search process, it turned into exploitative algorithm but sudden spike in diversity made HHO highly explorative search behavior. However, this did not maintain longer and HHO lost exploration ability to converge in sub-optimal locations, as compared to LMHHO. Contrarily, from exploration-exploitation graphs for low and high-dimensional problems, it shows that despite starting search with high exploration and low exploitation, LMHHO maintained exploration up to certain level and exploitation relatively lower than in HHO, after halfway towards the end of iterations. This confirms that maintaining exploration ability up to certain level towards the end of search process helps avoid ending up in losing even better optimal results. Fig. 20 provides summary of explorative and exploitative behaviors HHO and LMHHO on ten classic benchmark functions for $D = 30$ and $D = 500$.

Based on evidence of improvement in search behavior from the in-depth behavioral analysis presented in this study, following important theoretical characteristics can be supported from practical results provided in the form of diversity measurement and exploration-exploitation analysis:

- HHO is an efficient search strategy, however it loses population diversity towards the end of search process,
TABLE 10: Coefficients for IEEE-30 bus system

<table>
<thead>
<tr>
<th>Variables</th>
<th>( a )</th>
<th>( b )</th>
<th>( c )</th>
<th>( d )</th>
<th>( e )</th>
<th>( \beta )</th>
<th>( \gamma )</th>
<th>( w )</th>
<th>( \mu )</th>
</tr>
</thead>
<tbody>
<tr>
<td>G1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0.00375</td>
<td>18</td>
<td>0.037</td>
<td>4.991</td>
<td>-5.554</td>
<td>6.349</td>
</tr>
<tr>
<td>G2</td>
<td>2</td>
<td>0</td>
<td>1.75</td>
<td>0.00175</td>
<td>16</td>
<td>0.038</td>
<td>2.545</td>
<td>-6.047</td>
<td>5.638</td>
</tr>
<tr>
<td>G3</td>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0.00257</td>
<td>14</td>
<td>0.044</td>
<td>4.258</td>
<td>-5.094</td>
<td>4.586</td>
</tr>
<tr>
<td>G4</td>
<td>8</td>
<td>0</td>
<td>3.25</td>
<td>0.00834</td>
<td>12</td>
<td>0.045</td>
<td>5.326</td>
<td>-3.55</td>
<td>3.38</td>
</tr>
<tr>
<td>G5</td>
<td>11</td>
<td>0</td>
<td>3</td>
<td>0.025</td>
<td>13.5</td>
<td>0.041</td>
<td>6.131</td>
<td>-5.555</td>
<td>5.151</td>
</tr>
<tr>
<td>G6</td>
<td>13</td>
<td>0</td>
<td>3</td>
<td>0.025</td>
<td>13.5</td>
<td>0.041</td>
<td>6.131</td>
<td>-5.555</td>
<td>5.151</td>
</tr>
</tbody>
</table>

TABLE 11: OPF solutions obtained for IEEE-30 bus system

<table>
<thead>
<tr>
<th>Case</th>
<th>Variables</th>
<th>Min</th>
<th>Max</th>
<th>MGOA Case 1</th>
<th>MGOA Case 2</th>
<th>MGOA Case 3</th>
<th>IMFO Case 1</th>
<th>IMFO Case 2</th>
<th>IMFO Case 3</th>
<th>LMMHO Case 1</th>
<th>LMMHO Case 2</th>
<th>LMMHO Case 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>Fuel cost ($/h)</td>
<td>800.374</td>
<td>800.786</td>
<td>800.566</td>
<td>800.5912</td>
<td>800.3848</td>
<td>800.3848</td>
<td>800.3848</td>
<td>800.3848</td>
<td>800.3848</td>
<td>800.3848</td>
<td></td>
</tr>
<tr>
<td>Case 2</td>
<td>Emission (ton/h)</td>
<td>0.3649</td>
<td>0.3678</td>
<td>0.3719</td>
<td>0.3566</td>
<td>0.3653</td>
<td>0.3638</td>
<td>0.35824</td>
<td>0.3389</td>
<td>0.20278</td>
<td>0.2071</td>
<td></td>
</tr>
</tbody>
</table>

TABLE 12: OPF solutions comparison

<table>
<thead>
<tr>
<th>Objective functions</th>
<th>MGOA</th>
<th>GAOA</th>
<th>GA</th>
<th>PSO</th>
<th>TLBO</th>
<th>IMFO</th>
<th>LMMHO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>800.374</td>
<td>800.786</td>
<td>800.566</td>
<td>800.5912</td>
<td>800.3848</td>
<td>800.3848</td>
<td>800.3848</td>
</tr>
<tr>
<td>Case 2</td>
<td>0.3649</td>
<td>0.3678</td>
<td>0.3719</td>
<td>0.3566</td>
<td>0.3653</td>
<td>0.3638</td>
<td>0.35824</td>
</tr>
</tbody>
</table>

which hurts its ability of finding even better optimal locations.
- HHO maintains dynamic search behavior, as it regains exploration ability during search process, but suddenly keeps on losing population diversity in later part of iterations.

The long-term memory concept proposed in LMMHO provides ample options for population individuals to keep on searching for promising regions for even better optimal results.
- The proposed LMMHO also resembles HHO search behavior in initial search iterations, however opposite to HHO, it maintains exploitative ability up to a reasonable level to avail possibility of improving search results. The proposed long-term memory concept is general and can be implemented in various other population-based metaheuristic algorithms, for improved search efficiency.

VI. CONCLUSION

This study proposed long-term memory concept in HHO algorithm, to present modified variant in the form of LMMHO. The proposed LMMHO improved search efficiency by applying multiple global best locations in position update strategy. Instead of relying on single global best location, LMMHO keeps a record of multiple best experiences during search, which it utilizes to maintain population diversity even towards the end of iterations. The empirical analyses suggest that LMMHO did not lose exploration ability throughout iterations, and maintained it up to certain level. Despite, low
diversity as compared to HHO, LMHHO maintained trade-off balance between exploration and exploitation throughout towards the end of search process. While investigated on numerical optimization problems with low and high-dimensional landscapes, LMHHO proved its search efficiency by achieving better search results as compared to the original and various other established and recently introduced counterparts. LMHHO achieved global optimum value on eight out of ten 30 and 500-dimensional classic benchmark problems. These problems included both unimodal and multimodal functions. Moreover, LMHHO also generated superior results than the competitive methods on more than 68% of 100-dimensional CEC’17 functions. However, the complex functions of CEC’17 demand further improvement in LMMHHO, which extends opportunity for further research on developing more efficient methods for long-term memory related operations in LMHHO. Further evident from implementation on real-life optimization problem in the form of optimal power flow in power generation systems, LMHHO outperformed HHO and different other metaheuristic techniques from literature applied in the same problem. In OPF problem, LMHHO was employed on minimizing fuel cost, emission, and power loss when simulated on IEEE-30 bus system as test environment. Overall, the empirical study using both the categories of optimization problems affirmed that efficiency, stability, and applicability of the proposed approach.

Generally, from this study, it is learnt that when applying appropriate tools for in-depth performance analysis, a significant comprehension can be achieved on search behaviors of metaheuristic algorithms. As discussed in this paper, in the presence of numerous metaheuristic methods, with various different metaphors, the introduction of yet another algorithm may reinvent the wheel. Therefore, better analytical techniques, such as proposed in this study, are needed to make simple but effective modification in existing methods for improved search results. Additionally, there are also several potential research directions that can be drawn from this study. For example: (a) implementation of LMHHO on even higher dimensional problems with complex search space; (b) integration of the proposed long-term memory mechanism on various other metaheuristic algorithms; and (c) more item selection mechanisms, other than Roulette Wheel approach used for memory item selection in this paper, can be evaluated for developing robust search strategies.

REFERENCES


KASHIF HUSSAIN received the B.S. degree in information technology from the Mehran University of Engineering and Technology, Jamshoro, Pakistan, in 2006. Then he spent seven years in industry as Software Engineer for several software development companies in Karachi, Pakistan. He moved to Malaysia for further education, where he received M.S. and PhD. degrees in information technology from the Universiti Tun Hussein Onn Malaysia, Johor, Malaysia, in 2015 and 2018, respectively. From 2014 to 2018, he served as Research Assistant with Universiti Tun Hussein Onn Malaysia. Since 2019, he has been serving at Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu, China, as Postdoctoral Researcher. His major research interests include machine learning, optimization, fuzzy neural networks, and metaheuristic algorithms.

WILLIAM ZHU received the MS degree in mathematics from Xinjiang University, Wulumuqi, Xinjiang, China, in 1986 and the MS degree in systems engineering from the University of Arizona, Tucson, Arizona, in 2002. He worked at a university in China from 1986 to 2001 and was promoted to an associate professor in computer science in 1999. Since 2000, he has been collaborating with Dr. Fei-Yue Wang from the Institute of Automation, the Chinese Academy of Sciences, Beijing.

He studied as a PhD student at Northwestern Polytechnical University, Xi’an, China, from 1998 to 2001, at the University of Arizona from 2001 to 2003, and at the University of Auckland, Auckland, New Zealand, from 2003 to 2006. He is currently a Professor at the Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu, China. His research interests include granular computing, rough set, data mining, and artificial intelligence. He was listed in Highly Cited Researchers in computer science by Thomson Reuters in 2015, 2016 and 2017.

MOHD NAJIB MOHD SALLEH received the Ph.D. degree in computer science from the University of La Rochelle, France in 2008. He is currently a Professor at the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia. His research interests include decision support system, machine learning, fuzzy systems, and metaheuristic algorithms.