Wild Horse Optimizer-Based Spiral Updating for Feature Selection

AHMED A. EWEEs1,2, (Senior Member, IEEE), FATMA H. ISMAIL3, AND RANIA M. GHONIEM4

1Department of Information Systems, College of Computing and Information Technology, University of Bisha, Bisha 61922, Saudi Arabia
2Department of Computer, Damietta University, Damietta 34517, Egypt
3Faculty of Computer Science, Misr International University, Cairo 11341, Egypt
4Department of Information Technology, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Riyadh 11671, Saudi Arabia

Corresponding author: Ahmed A. Ewees (ewees@du.edu.eg)

This work was supported by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R138), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

ABSTRACT Feature selection (FS) is a vital and challenging process in several domains, including data mining, data clustering, text mining, education, biology, medicine, public health, machine learning, image processing, and so on. The greedy and comprehensive algorithm methods cannot identify the best subset amid the rising number of features. Thus, swarm-based algorithms are becoming more popular for identifying the best group of features. This study relies on the spiral-updated position of the Whale Optimization Algorithm (WOA) to propose an improved version of the Wild Horse Optimizer (WHO). This improvement enhances the WHO’s ability to update solutions and explore various possibilities in the search domain. The proposed method (WHOW) was assessed using two experiments to confirm the efficacy of the improved optimizer. The first experiment was global optimization using CEC 2019 benchmark functions, whereas the second was an FS by testing 20 benchmark datasets. The results obtained using the proposed WHOW method were compared to some popular algorithms over the benchmark datasets of global optimization and FS. The experimental results reflect the superiority of WHOW in the solutions to different optimization problems and its ability to select prominent features over most benchmark datasets. These results are due to the implementation of bubble nets in the WHO using spiral movement, which promotes flexibility and performance.

INDEX TERMS Wild horse optimizer, whale optimization algorithm, spiral updating position, global optimization, feature selection.

I. INTRODUCTION Dimensionality reduction is the primary concern of many data mining algorithms, particularly classification tasks. The massive number of features in a dataset badly affects the performance of classifiers since many of them might be unimportant [1]. Therefore, feature selection (FS) is a mandatory preprocessing step [2]. Two methods are used for the FS task: filter methods [3], and wrapper methods [4]. This study focuses on wrapper methods, which depend on the selection and evaluation of a subset from the features. Although the wrapper methods are accurate, they are computationally exhaustive. They rely on selecting a subset of features and calculating the accuracy of the applied classifier.

The goal is to select the minimum number of features that yields the highest classification accuracy. In this regard, Swarm-based algorithms are perfect candidates for selecting the best features to represent the dataset. These algorithms, which are divided into two phases, mimic the behavior of swarms in their search for a source of food. In the exploration phase of the swarm technique, the population searches the entire problem space. In the exploitation phase, the populations improve their positions iteratively to solve the problem. A balance between these two phases is a mandatory criterion to avoid sticking to local optima. However, as noted in the no free lunch (NFL) theory [5] not all
metaheuristic algorithms can solve all problems. For this reason, researchers implemented many metaheuristic algorithms that can be classified into gradient-based methods (GB) and non-gradient-based methods. The Gradient-based (GB) methods include; a gradient-based optimizer [6]; Newton’s method [7], and Levenberg Marquardt’s algorithm [8] depending on the derivative of the objective function. Thus, they converge slowly but with no guarantee of a successful solution to various optimization problems [9], [10], [11], [12]. Non-GB (MH) methods, on the other hand, start at random positions, and the search direction is updated by exploration and exploitation. MH algorithms are robust at finding global optimum solutions but require huge processing power. Various MH algorithms have inherent binary variants to solve the FS problem, such as the particle swarm optimization (PSO) [13], [14], [15], artificial ant colony (ACO) [16], bee colony algorithm (BCA) [17], ant lion optimizer (ALO) [18], salp swarm algorithm (SSA) [19], [20], grasshopper optimization algorithm [21], and moth-flame optimization (MFO) [22], [23].

As mentioned earlier, the origins of metaheuristic algorithms differ. While some are found in biology, others are found in physics. However, the authors in [24] introduced the league championship algorithm based on sports, and some researchers in [25] applied chaotic maps to improve its convergence rate. The law of light reflection has inspired the researchers in [26] to develop a new MH called optical-inspired optimization (OIO) that borrows the concept of convergence and divergence of mirrors. In this algorithm, the surface of the optimized function is a wavy mirror whose peaks and valleys are concave and convex lenses. The populations in OIO are artificial light points whose rays are reflected on the function surface to determine whether it is a minimum or maximum point. The classical OIO is improved in [27] through the addition of chaos theory, whereby chaotic maps improved the global convergence rate of the OIO and prevented ticking in local optima. However, the classical OIO achieved good results in solving the traveling tournament problem and minimized its parameters, the total movement of teams, and transportation [28]. Light-based optimization algorithms are assessed and proven as efficient relative to other competitive algorithms in solving specific problems [29].

There is NFL theory [5] suggests that there is no optimization algorithm that can solve all problems. That explains the need to devise many algorithms to cope with all emerging problems. The study presented in [30] enumerated many advantages and disadvantages of metaheuristic algorithms. The shortfall ranges from slow convergence, sticking into local optima, to a shortage in search space coverage. Those drawbacks provoked the researchers to hybridize two metaheuristic algorithms or even devise new algorithms. This study addressed the low-search space coverage drawback of the WHO.

Thus, the study contributes to the literature by proposing a novel version of the WHO algorithm utilizing the spiral-updating position strategy. The proposed algorithm is evaluated using both global optimization and FS problems.

The rest of the paper is organized as follows: Section II presents some of the related works. Section III describes the methods and materials. Section IV introduces the proposed methods. Section V evaluates the proposed methods and discusses the results. The conclusions are presented in Section VI.

II. RELATED WORK

The Wild Horse Optimizer (WHO) algorithm is relatively new and has little application in the literature for FS purposes. However, many authors [31] interested in increasing the efficiency of photovoltaic applications have used WHO for parameter optimization in their problem domain. The parameters generated by the wild horse were more accurate than those of any other algorithms. WHO has been widely used to increase the reliability and stability of optimal radial distribution network systems [32]. Therefore, WHO is useful for optimizing the parameters in many problem domains. The study of [33] proposed an improved version of WHO by generating new candidate solutions using the cuckoo search algorithm. This improvement accelerated the WHO convergence speed. They evaluated their approach on benchmark functions, where the results showed the superiority of the improved WHO.

The classical WHO is used to optimize the parameters of an application in a parallel series system known as the reliability redundancy allocation problem [34]. An improved version of the WHO is introduced in [35] to solve global optimization problems. The authors used three operators to boost the exploitation behavior of WHO leading to a balance between the exploitation and exploration phases. The whale optimization algorithm (WOA) is useful in the FS research area. The authors in [36] introduced an approach based on WOA and differential evolution operators (DE) [37] to overcome sticking in local optima. Their approach relies on an elite opposition-based learning technique to enhance the initialization phase of WOA, and DE operators are used at the end of each iteration of WOA. The authors compared their approach with other optimization algorithms and two deep learning techniques in Arabic sentiment analysis FS. The results showed that the hybrid WOA achieved the best accuracy and fewer features. In [38], the quantum concept is unified with WOA FS. The quantum bit individual representation and the quantum gate variation operator enhanced the exploration and exploitation of the WOA. The traditional WOA was compared with quantum-based WOA, and the latter was applied for a wrapper FS. However, quantum-based WOA improved the average classification accuracy, average fitness, and average area under the curve. The continuous position of whales is converted into a binary correspondent using the eight different transfer functions in [39], [40], and [41]. In some applications, the V-shaped transfer function achieved the best performance over the S-shaped one, whereas it is different in other applications. Many researchers, such as [42],
focus on enriching the application field of WOA and treating its shortcomings. The shortfall of WOA is summarized as slow convergence and unbalance between global and local search capabilities. The proposed approach in [42] improved the exploration capability of WOA through a hybrid mutation strategy based on the Gaussian and Cauchy mutation operators. The added operators addressed the random learning problem of the WOA exploration phase. However, the improved WOA achieved high accuracy and less number of features. The flower pollination algorithm (FPA) [43] is combined with WOA and opposition-based learning (OBL) to offer a new hybrid approach to wrapper FS methods [44]. FPA improves the WOA solutions with two global and local search processes in an opposite space to the solutions of the WOA, while OBL ensures the convergence and accuracy of the algorithm. Their proposed approach has been applied in spam e-mails detection and compared with other algorithms to prove its highest classification accuracy.

The Sine cosine algorithm (SCA) [45] is combined with WOA and a chaotic logistic map to produce a new enhanced version of WOA [46]. SCA improves the exploitation process of the traditional WOA, which prevents it from sinking in the optimal local area of the solution. Similarly, using chaotic maps improves the exploration phase and the diversity of populations. The new WOA is converted to a binary variant using S-shaped and V-shaped transfer functions to act as a wrapper FS algorithm. The new proposed WOA proved to be superior to other algorithms when applied to the problem of predicting students’ performance. Although WHO is a new promising meta-heuristics, it suffers from an imbalance between exploitation and exploration. This shortage affects the algorithm convergence speed. However, many past and recent studies proved the efficiency of WOA in global optimization tasks.

III. METHODS

This section briefly explains the WHO and Whale Optimization algorithms.

A. WHO

The WHO [47] mimics the behavior of wild horses. Wild horses are called non-terrestrial horses. They live in two groups, one for female horses (mares), a family group, and the other is a single group for male horses (stallions). Mating happens between the family group and the single group. At the beginning of their life, foals (child horses) care about grazing. Female foals leave their family group and join others. Male foals are known as stallions when they reach puberty. Decency, in the sense that gathering the stallions in a single group prevents incest. Striving for water in dry seasons showed the act of dominant leaders who can access water holes while other low-dominant members wait for hours. Mares lead family groups; however, mares are subordinate and must follow a leader selected from the stallions. The main steps of the WHO are as follows.

1) POPULATION INITIALIZATION AND LEADERSHIP SELECTION

The Initial population (\(\hat{x}\)) with \(N\) members is randomized such that (\(\hat{x}\) = \(\{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n\}\)) and the objective function of each population is calculated to form the following vector

\[
\hat{O} = \{O_1, O_2, \ldots, O_n\}
\] (1)

The population is divided into groups \(G\) where \(G = NXPS\), where \(PS\) denotes the percentage of stallions within the whole populations. Each group has a leader from stallions, randomly initialized, but in later stages of the algorithm, the highest fitness value controls the selection of leaders.

2) GRAZING BEHAVIOR

The grazing behavior is shown in Equation 2.

\[
\hat{x}_{i,G}^j = 2Z \cos(2\pi RZ) \times \left( \text{Stallion}^j - \hat{x}_{i,G}^j \right) + \text{Stallion}^j
\] (2)

where \(\hat{x}_{i,G}^j\) is the current position of a member’s group, \(\text{Stallion}^j\) indicates the position for the leader’s group, the \(Z\) parameter is given as shown in Equation 3, \(R\) is a uniformly distributed random number in the range \([-2, 2]\) that results in horses grazing at various angles (360 degrees) of the group leader. \(\pi\) is taken as 3.14, cosine function of \(R\) and \(\pi\) causes the movement in different radii, and the final position of a member \(\hat{x}_{i,G}^j\) is the updated position of a member.

\[
P = \bar{R}_1 < TDR; \quadIDX = (P == 0);
Z = R_2 \Theta \bar{IDX} + \bar{R}_3 \Theta (\sim \bar{IDX})
\] (3)

where \(P\) is a vector \(\in [0, 1]\), \(\bar{R}_1\) and \(\bar{R}_3\) are random vectors \(\in [0, 1]\), \(R_2\) defines a random number \(\in [0, 1]\), \(IDX\) indexes of the \(\bar{R}_1\) returns that satisfy the condition \((P == 0)\). \(TDR\) decreases from 1 to 0 as in Equation 4.

\[
TDR = 1 - \text{iter} \times \left( \frac{1}{\text{maxiter}} \right)
\] (4)

where \text{iter} denotes the current iteration and \text{maxiter} denotes the maximum iterations’ number.

3) HORSE MATING BEHAVIOR

Decency and mating behavior are presented in Equation 5.

\[
X_{G,k}^p = \text{Crossover} \left( X_{G,j}^q, X_{G,q}^i \right), \quad i \neq j \neq k \quad p = q = \text{end},
\]

\[
\text{Crossover} = \text{Mean}
\] (5)

where \(X_{G,k}^p\) denotes the position of horse \(p\) as it leaves group \(k\) and a horse whose parents leave groups \(i\) and \(j\) due to puberty takes it replaces it. They are not related to one other and have mated, and reproduced. \(X_{G,j}^i\) The position of the foal \(q\) belongs to \(i\) group and after reaching the age of puberty, it mated with the horse \(z\) with the position \(X_{G,z}^j\) which leaves \(j\) group.
B. GROUP LEADERSHIP
The leader’s group must direct the group to an appropriate location of the water. Leaders struggle for this water for the domination group to use, and the others are not permitted to use it until the domination group leaves. Equation 6 shows this behavior, as in (6), shown at the bottom of the next page, where Stallion \( G_i \) is the leader’s next position of the \( i \) group, WH is the water position, Stallion \( G_i \) is the current leader’s position of the group \( i \).

1) EXCHANGE AND LEADERSHIP SELECTION
The leaders are selected randomly at the beginning. However, the fittest population is selected as the leader at a later stage of the algorithm. The positions of the leader and the selected member are shown in (7), as shown at the bottom of the next page.

C. WHALE OPTIMIZATION ALGORITHM
The WOA was introduced by [48]. WOA mimics the foraging behavior of humpback whales, where they encircle their prey using a net of bubbles called bubble-net feeding behavior. The steps of WOA can be described as follows:

1) INITIALIZATION AND ENCIRCLING PREY BEHAVIOR
The initial population \((\bar{x})\) with \( N \) members is randomized such that \((\bar{x}) = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)\) and the objective function of each search agent is calculated and the best one is represented as \( X^* \).

This behavior simulates the encircling of prey by a set of humpback whales. After defining the position of the best search agent, other agents update their positions accordingly, as shown in Equation 8.

\[
\bar{D} = \left| \bar{C} \cdot \bar{X}^*(t) - \bar{X}(t) \right|
\]

\[
X(t+1) = \bar{X}^*(t) - A \cdot D
\]  

(8)

where \( t \) denotes the current iteration, \( A \) and \( \bar{C} \) are two vectors described in equations 9-10. \( \bar{X}^* \) denotes the best search agent obtained in iteration \( t \) and \( \cdot \) (dot) is the element multiplication and \( \mid \) is the absolute value. In equations 9 and 10, \( \bar{a} \) is a vector decreasing from 2 to 0 during iterations, and \( \bar{r} \) is a random vector in the range of \([0,1]\).

\[
\bar{A} = 2\bar{a} \cdot \bar{r} - \bar{a}
\]

\[
\bar{C} = 2 \cdot \bar{r}
\]  

(9) (10)

2) EXPLOITATION PHASE
Bubble-net-attacking behavior is divided into the following stages:

- Shrinking encircling stage: As humpback whales encircle prey, a shrinking action is achieved by decreasing the value of \( \bar{a} \) from 2 to 0 over all iterations in Equation 9.
- Spiral-updating position stage: The distance between the prey positioned in \((X^*, Y^*)\) and the whale positioned in \((X,Y)\) is calculated in this stage, as shown in Equation 11. The helix shape movement is presented in Equation 12.

\[
\bar{X}(t+1) = \bar{D}^l \cdot 2l \cdot \cos(2\pi l) + \bar{X}^*(t)
\]

\[
\bar{X}(t+1) = \begin{cases} 
\bar{X}^*(t) - \bar{A} \cdot \bar{D} & \text{if } p < 0.5 \\
\bar{D}^l \cdot 2l \cdot \cos(2\pi l) + \bar{X}^*(t) & \text{if } p \geq 0.5 
\end{cases}
\]  

(11) (12)

where \( \bar{D}^l = \left| \bar{X}^*(t) - \bar{X}(t) \right| \) is the distance between \( i \)th whale and the best whale position, \( b \) is a constant that defines the shape of the spiral logarithm, \( r \) is a random number in the range \([-1,1]\). \( p \) is the probability that chooses the mechanism of update.

Algorithm 1 WHO Pseudo-Code

1: Initialization: initialize the parameters \( PC, PS \).
2: Initialize populations.
3: Calculate the fitness value of each population
4: Form groups and select leaders.
5: while \((\text{iter} \leq \maxiter)\) do
6: calculate TDR as in Equation 4.
7: for each stallion do
8: calculate \( Z \) as in Equation 3.
9: for each foal inside the group do
10: if \( \text{rand} > PC \) then
11: update position by Equation 2
12: else
13: update position by Equation 5
14: end if
15: end for
16: if \( \text{rand} > 0.5 \) then
17: update position of Stallion \( G_i \) by Equation 6
18: end if
19: else
20: update position of Stallion \( G_i \) by Equation 6
21: if \( \text{fitness(Stallion)} > \text{fitness(Stallion)} \) then
22: Stallion = Stallion \( G_i \)
23: end if
24: end if
25: end for
26: select the foal with minimum fitness
27: if \( \text{fitness(foal)} < \text{fitness(Stallion)} \) then
28: exchange foal and stallion position according to Equation 7
29: end if
30: \( m = m + 1 \)
31: end while
32: Return the solution with the best fitness
3) EXPLORATION PHASE
Contrary to the exploitation phase, the whale position is updated based on the position of a random whale, as shown in Equation 13.

\[
\vec{D} = |\vec{C} \cdot \vec{X}_{\text{rand}} - \vec{X}|
\]

\[
X(t + 1) = \vec{X}_{\text{rand}} - \vec{A} \cdot \vec{D}
\]

where $|\vec{A}| > 1$.

**Algorithm 2** WOA Pseudo Code

1. Initialize whale populations.
2. Calculate the fitness values and select the population with the best fitness value as $X^*$
3. while $(t <= \text{maxiter})$ do
4. for each search agent do
5. Update $p,A,C,A,a$. 
6. if $p < 0.5$ then
7. if $|A| < 1$ then Update position by the first part of Equation 12
8. else if $|A| >= 1$ then Select a random population $X_{\text{rand}}$ and update position by Equation 13
9. end if
10. else
11. end if
12. Update the position of population using the second part of Equation 12
13. end if
14. Amend any population that goes outside the search space
15. Calculate the fitness of each population
16. Update $X^*$
17. $t = t + 1$
18. end while
19. Return the solution with the best fitness $X^*$

IV. PROPOSED METHOD
This section presents the descriptions of the proposed method WHOW. The proposed WHOW improves the optimization behavior of the WHO using the spiral-updating position of the WOA. This improvement enhances the ability of the WHO to update the solutions and explore various possibilities in the search domain. The “spiral-updating position” stage increases the exploration behavior of the proposed algorithm, thereby increasing its ability to explore more regions in the search space. This step helps the standard WHO to effectively extend the search for global optima in the problem domain. Additionally, calculating the distance between the whale and the prey plays a good role in guiding the WHOW to find the best area in the search space mimicking the helix movement of the whale.

In this regard, the “spiral-updating position” is applied based on probability ($p$) which is calculated using Equation 14 to determine which operators could be used with the current solution. Therefore, if $p < 0.5$, the “spiral-updating position” is applied, or else the operators of WHO are used.

\[
p = \frac{f_i}{\sum_{i=1}^{N} f_i}
\]

where, $f_i$ is the value of the fitness function.

The first step in the application of WHOW is to determine the values of all parameters and generate the search domain for the initial population. Secondly, to select the best solution, the quality of each solution in the population is evaluated using the fitness function, as in Equation 15.

\[
f(x) = \gamma E_x + (1 - \alpha)(\frac{1}{x}(||C||))
\]

where $\gamma E_x$ is the error of the classification process, ($K$-NN classifier is used in this study). The second part defines the ratio of the selected feature. $\alpha \in [0,1]$ balances the classification error and the number of selected features.

Thirdly, the processing of the optimization steps starts by exploring the search domain to find the initial optimal solution by evaluating the current population using the fitness function. Finally, the values obtained through the fitness function are computed and the best one is determined and saved. The sequence of the WHOW is presented as follows:

- Define the values of the global parameters for the experiment.
- Create the population $X$ with $D$ dimension and $N$ size.
- Compute the fitness values for all solutions, and then the main iterations of the WHOW are started.

This sequence is iterated till it reaches the stop condition. The full structure of the WHOW is illustrated in Figure 1.

\[
\text{Stallion}_{G_i} = \begin{cases} 2C \cos(2\pi RZ) \times (\text{WH} - \text{Stallion}_{G_i}) + \text{WH} & \text{if } R_3 > 0.5 \\ 2C \cos(2\pi RZ) \times (\text{WH} - \text{Stallion}_{G_i}) - \text{WH} & \text{if } R_3 \leq 0.5 \end{cases}
\]

\[
\text{Stallion}_{G_i} = \begin{cases} X_{G,i} & \text{if } \cos t (X_{G,i}) < \cos t (\text{Stallion}_{G_i}) \\ \text{Stallion}_{G_i} & \text{if } \cos t (X_{G,i}) > \cos t (\text{Stallion}_{G_i}) \end{cases}
\]
The complexity of the WHOW depends on the dimension of the problem \((D)\), the size of the population size \((N)\), and the number of iterations \((\text{maxiter})\) of the WHO and the parameters of the WOA depend on a probability value and it does not add more complexity. Therefore, the complexity of the WHOW in the worst case is as follows: \(O(N \times D \times \text{maxiter} + N \times D + N \times \text{maxiter})\).

V. EXPERIMENTS AND DISCUSSION

In this section, the proposed WHOW is evaluated using two experiments; the first one is global optimization based on CEC2019 benchmark functions, whereas the second one is the FS using 20 benchmark datasets. The proposed method is then compared with WHO [47], PSO [49], GA [50], SCA [45], MVO [51], SSA [19], SMA [52], MPA [53], and WOA [48]. In the experiments, 100 iterations were performed on a population size of 20, and 20000 function evaluations were applied as a stopping criterion. For statistical purposes, all methods are applied in 30 runs. The experiment used “Matlab 2018b” on “Windows 10” over a “Core i5 CPU” and “8 GB RAM”. Table 1 shows the parameters of the compared algorithms.

To evaluate the performance of the proposed method, the following performance measures are used in the experiments: the maximum (Max) and minimum (Min) of the fitness function as in equations 17 and 16, respectively, and the standard
Algorithm 3 Proposed WHOWOA Pseudo Code

1: Initialization: initialize the parameters of WHO and WOA.
2: Initialize populations.
3: Calculate the fitness values.
4: Form groups and select leaders.
5: while (iter <= maxiter) do
6:   for each stallion do
7:     calculate TDR as in Equation 4.
8:     for each foal inside the group do
9:        if rand > PC then
10:           if p < 0.5 then
11:              Updating by Equation 11
12:           else
13:              Updating by Equation 2
14:        end if
15:     else
16:        Update position by Equation 5
17:     end if
18:   end for
19:   if rand > 0.5 then
20:     Updating Stallion \( G_i \) by Equation 6 first part
21:   else
22:     Updating Stallion \( G_i \) by Equation 6 second part
23:   end if
24:   if \( \text{fitness}(\text{Stallion}_{G_i}) > \text{fitness}(\text{Stallion}) \) then
25:     Stallion = Stallion \( G_i \)
26: end if
27: end while
28: Return the solution with the best fitness

These measures are selected because they are used in the state-of-the-art and have an excellent ability to compare the results of the algorithms from several perspectives.

- Minimum fitness value (Min): It is applied for determining the best fitness value (the smallest in terms of problem minimization) realized in a particular algorithm, this measure is used as follows:

\[
Min = \min_{1 \leq i \leq N} \text{fit}_i
\]  

(16)

- Maximum fitness value (Max): It is used for determining the worst fitness value (the greatest in terms of problem minimization) realized in a particular algorithm, this measure is used as follows:

\[
\text{Max} = \max_{1 \leq i \leq N} \text{fit}_i
\]  

(17)

- Standard deviation (STD): It is used for measuring the extent of the dispersion of the fitness values from the average (\( \mu \)) value. The lower the STD, the more robust the method. This measure is used as follows:

\[
\text{STD} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\text{fit}_i - \mu)^2}
\]  

(18)

where, \( \text{fit} \) and \( \mu \) are the value and mean of the fitness function, respectively, \( N \) is the sample size.

Additionally, there are some measures for the FS experiment, namely, the classification accuracy (Acc) as in Equation 19 and the number of selected features by each method. The Acc is applied to measure the ratio of correct
The standard deviation results over the 10 test functions are provided in Table 4. As seen in the table, the WHOW is the most stable metaheuristic among the algorithms, achieving the smallest values in 8 functions. The MPA and WHO algorithms rank second and third in terms of stability. In contrast, the PSO is the least stable algorithm, ranking last among all competitor algorithms.

In terms of convergence speed, as shown in Table 5, the SCA algorithm takes the shortest time to converge, followed by the PSO and SSA algorithms. The WHOW ranked fourth, achieving optimal results in three functions (F2, F3, and F10). The SMA, GA, and MVO algorithms are the slowest in terms of convergence. Moreover, Figure 2 illustrates the convergence curves for all methods, and it is clear from the Figure that the proposed WHOW showed the best convergence behavior in most functions.

The minimum values given by each metaheuristic algorithm are shown in Table 6. The Table showed that WHOW and WHO each showed the most optimal values in 3 functions (F2, F3, and F5 for WHOW) and (F1, F3, and F6 for WHO). The MVO and SSA algorithms ranked second since both showed the best values in 2 functions (F7 and F9 for MVO) and (F4 and F6 for SSA). The SMA and MPA algorithms showed optimal values in only one function.
function (F1 and F2, respectively) and thus rank third among the competitor algorithms. The SCA algorithm obtained the lowest rank among all competitor algorithms since it failed to reach optimal values in 6 functions (F4-F6 and F8-F10).

The maximum values taken by each competitor algorithm are shown in Table 6. The WHOW ranked first, showing optimal values in 7 of 10 functions (F2-F5, F7, and F9-F10). The WHO algorithm came second, showing optimal values in 3 functions (F2-F3 and F10). The MPA algorithm ranked
third since it reached the best values in four functions (F2-F4 and F10). The SCA algorithm obtained the lowest rank as it failed to reach optimal values in 5 functions (F6-F10).

Table 8 presents the results of the statistical test of Wilcoxon’s rank-sum (i.e., \(P - value\)) to examine the significant differences among the WHOW and its competitors in the CEC2019 benchmark. Table 8 shows that there is a significant difference between WHOW and the other algorithms in most functions of the CEC2019 benchmark. The results of the Friedman test are shown in Table 9. Similarly, it is shown that the WHOW outperformed the other algorithms and ranked first in 7 of 10 functions, followed by MPA and WHO (see Table 9).

**B. SOLVING FS PROBLEMS**

This section evaluates the proposed WHOW for solving twenty benchmark datasets [55], the description is presented in Table 10.
TABLE 8. Results of the Wilcoxon rank-sum test.

|       | WHO | PSO | GA | SCA | MVO | SSA | SMA | MPA | WOA |
|-------|-----|-----|----|-----|-----|-----|-----|-----|-----|
| F1    | 0.67198 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.38559 | 0.00000 | 0.00000 |
| F2    | 0.33789 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.17080 | 0.33789 |
| F3    | NaN  | 0.00000 | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| F4    | 0.00006 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.05487 | 0.04503 |
| F5    | 0.65361 | 0.00138 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.47796 | 0.00000 |
| F6    | 0.00002 | 0.21524 | 0.00000 | 0.00000 | 0.00000 | 0.23548 | 0.00000 | 0.00000 | 0.00000 |
| F7    | 0.16598 | 0.42265 | 0.00002 | 0.00000 | 0.00010 | 0.00228 | 0.00007 | 0.00220 | 0.00000 |
| F8    | 0.00299 | 0.03655 | 0.00000 | 0.00000 | 0.00554 | 0.00003 | 0.00000 | 0.02141 | 0.00000 |
| F9    | 0.04616 | 0.00000 | 0.00010 | 0.00000 | 0.06030 | 0.00000 | 0.00086 | 0.04612 | 0.00000 |
| F10   | 0.00142 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 | 0.00000 |

TABLE 9. Results of the Friedman test.

|       | WHO | PSO | GA | SCA | MVO | SSA | SMA | MPA | WOA |
|-------|-----|-----|----|-----|-----|-----|-----|-----|-----|
| F1    | 1.854 | 2.042 | 10.000 | 7.917 | 5.979 | 6.354 | 6.375 | 2.104 | 4.000 | 8.375 |
| F2    | 2.896 | 3.000 | 10.000 | 6.313 | 7.583 | 8.875 | 3.813 | 3.125 | 3.000 | 6.396 |
| F3    | 5.000 | 5.000 | 9.771 | 5.000 | 5.000 | 5.000 | 5.000 | 5.000 | 5.000 | 5.229 |
| F4    | 2.417 | 5.104 | 9.375 | 6.000 | 9.167 | 4.833 | 5.021 | 2.917 | 1.750 | 8.417 |
| F5    | 2.292 | 2.438 | 9.542 | 3.958 | 9.208 | 5.646 | 4.563 | 6.458 | 2.646 | 8.250 |
| F6    | 3.042 | 4.500 | 7.938 | 3.292 | 9.479 | 8.188 | 2.792 | 6.500 | 1.833 | 7.438 |
| F7    | 2.708 | 3.563 | 3.625 | 5.646 | 9.833 | 5.042 | 5.542 | 6.000 | 3.958 | 9.083 |
| F8    | 2.354 | 4.708 | 5.958 | 3.896 | 9.667 | 4.417 | 3.813 | 6.667 | 3.396 | 8.125 |
| F9    | 2.333 | 3.813 | 8.500 | 5.146 | 9.708 | 3.521 | 6.396 | 3.667 | 3.542 | 8.375 |
| F10   | 3.083 | 3.813 | 7.875 | 4.188 | 9.750 | 6.875 | 2.229 | 7.229 | 2.042 | 7.917 |

TABLE 10. Description of the used FS datasets.

| Name             | Instances | Features | Classes |
|------------------|-----------|----------|---------|
| breastWDBC      | 569       | 30       | 2       |
| ionosphereDS    | 351       | 34       | 2       |
| sonarDS         | 208       | 60       | 2       |
| glassDS         | 214       | 9        | 7       |
| tic-tac-toeDS   | 958       | 9        | 2       |
| VoteDS          | 300       | 16       | 2       |
| hearDS          | 270       | 13       | 2       |
| LymphographyDS  | 148       | 18       | 2       |
| waveformDS      | 5000      | 40       | 3       |
| cleandataDS     | 476       | 166      | 2       |
| ZooDS           | 101       | 16       | 6       |
| SPECTDS         | 267       | 22       | 2       |
| wineDS          | 178       | 13       | 3       |
| breastcancerDS  | 699       | 9        | 2       |
| ecoliDS         | 336       | 7        | 8       |
| CongressEWDS    | 435       | 16       | 2       |
| M-of-nDS        | 1000      | 13       | 2       |
| ExactlyDS       | 1000      | 13       | 2       |
| Exactly2DS      | 1000      | 13       | 2       |
| krvskpDS        | 3196      | 36       | 2       |

Considering the average fitness function values in Table 11, the WHOW showed the best values in 19 of 20 datasets, followed by the MPA algorithm, which provided the best results in 5 of 20 datasets (wineDS, tic-tac-toeDS, ZooDS, ExactlyDS, and M-of-nDS). The PSO and MVO ranked third as each provides optimal results over four datasets. The GA and SCA came in the fourth rank, performing well in three datasets. The SSA and WOA failed to return the best values in most tested datasets. Moreover, Figure 3 shows the average results for all datasets based on the fitness function values.
Considering the stability of the tested algorithms in Table 12, the WHOW algorithm is the most stable meta-heuristic because it has the lowest standard deviation in 9 (wineDS, breastcancerDS, LymphographyDS, tic-tac-toeDS, ZooDS, Exactly2DS, M-of-nDS, and krsvkpDS) out of the 20 benchmark datasets. The MPA ranked second since it showed stability in seven datasets. In contrast, the PSO and WOA algorithms ranked third, showing the lowest standard deviations in 4 of the 20 benchmark datasets. The SSA, SMA, and WHO algorithms were less stable, achieving minimum standard deviations in only three datasets.

Table 13 shows the minimum values of each algorithm in the datasets. From the table, WHO and PSO algorithms achieved optimal value in 14 of the 20 benchmark datasets. The MPA algorithm ranked second, achieving the best values in 13 datasets. The WHOW and GA algorithms came third, achieving the best values in 12 datasets. MVO and SSA ranked fourth and fifth, respectively. The SMA algorithm

| Fitness Function          | WHOW   | WHO    | PSO    | GA     | SCA    | MVO    | SSA    | SMA    | MPA    | WOA    |
|--------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| breastWDBCDS             | 0.08371| 0.09712| 0.10637| 0.11288| 0.10263| 0.13268| 0.13519| 0.30501| 0.11842| 0.14901|
| ionosphereDS             | 0.07367| 0.15311| 0.14178| 0.14178| 0.16797| 0.18830| 0.14682| 0.43519| 0.09517| 0.25254|
| wineDS                   | 0.00000| 0.04264| 0.00000| 0.00000| 0.07538| 0.03015| 0.35181| 0.27365| 0.00000| 0.03015|
| breastcancerDS           | 0.15753| 0.19691| 0.19499| 0.21486| 0.23501| 0.22058| 0.23132| 0.47226| 0.16765| 0.22058|
| glassDS                  | 0.80807| 0.83371| 0.83133| 0.81058| 0.93456| 0.87458| 1.09662| 1.06528| 0.87032| 0.87298|
| sonarDS                  | 0.04903| 0.13392| 0.08321| 0.09469| 0.12629| 0.10351| 0.20087| 0.53818| 0.06966| 0.16820|
| LymphographyDS           | 0.24991| 0.33289| 0.36313| 0.33345| 0.38414| 0.33330| 0.34390| 0.63209| 0.29941| 0.28744|
| tic-tac-toeDS            | 0.00000| 0.09656| 0.00000| 0.05010| 0.38801| 0.00000| 0.11927| 0.52923| 0.00000| 0.00000|
| waveformData              | 0.63670| 0.64209| 0.65957| 0.66035| 0.67922| 0.65703| 0.64633| 0.92329| 0.64223| 0.64712|
| clean1dataDS             | 0.15054| 0.21281| 0.15793| 0.16879| 0.19965| 0.21925| 0.22221| 0.44353| 0.15460| 0.27185|
| SPECDTS                  | 0.34500| 0.35720| 0.35785| 0.36209| 0.37274| 0.35740| 0.37379| 0.49172| 0.36651| 0.38913|
| ZooDS                    | 0.00000| 0.00000| 0.16000| 0.00000| 0.00000| 0.00000| 0.14928| 0.93075| 0.00000| 0.04000|
| ecoliDS                  | 1.19267| 1.43145| 1.37819| 1.44258| 1.41509| 1.38047| 2.45395| 2.03103| 1.41824| 1.72973|
| CongressEWDS             | 0.12323| 0.13081| 0.13353| 0.15277| 0.16351| 0.16514| 0.16058| 0.38948| 0.13874| 0.17460|
| ExactlyDS                | 0.48576| 0.48691| 0.50249| 0.49226| 0.47741| 0.49389| 0.51077| 0.59657| 0.47978| 0.48903|
| Exactly2DS               | 0.00000| 0.00000| 0.00000| 0.00000| 0.00000| 0.00000| 0.00000| 0.00000| 0.00000| 0.00000|
| M-of-nDS                 | 0.00000| 0.00000| 0.00000| 0.00000| 0.00000| 0.00000| 0.00000| 0.00000| 0.00000| 0.00000|
| VoteDS                   | 0.10887| 0.14928| 0.14194| 0.12841| 0.11925| 0.14194| 0.13575| 0.41685| 0.11151| 0.16841|
| krsvkpDS                 | 0.13381| 0.14446| 0.16029| 0.16432| 0.14234| 0.13990| 0.15218| 0.59988| 0.13535| 0.18550|
| heartDS                  | 0.32486| 0.33167| 0.32645| 0.33378| 0.37536| 0.35131| 0.38783| 0.50392| 0.33319| 0.38558|
TABLE 13. Minimum fitness function values of competitor algorithms in benchmark datasets.

| STD             | WHOW | WHO  | PSO  | GA   | SCA   | MVO   | SSA  | SMA  | MPA  | WOA  |
|-----------------|------|------|------|------|-------|-------|------|------|------|------|
| breastWDBCDS    | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.083918 | 0.0000 | 0.205557 | 0.083918 | 0.118678 |
| ionosphereDS     | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.106600 | 0.106600 | 0.0000 | 0.319801 | 0.0000 | 0.184637 |
| wineDS          | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.261116 | 0.0000 | 0.0000 | 0.0000 |
| breastcancerDS   | 0.150756 | 0.150756 | 0.106600 | 0.184637 | 0.184637 | 0.184637 | 0.184637 | 0.213201 | 0.106600 | 0.184637 |
| glassDS         | 0.764791 | 0.672927 | 0.672927 | 0.644278 | 0.752355 | 0.475831 | 0.900733 | 0.951662 | 0.752355 | 0.658758 |
| sonarDS         | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.392232 | 0.0000 | 0.0000 |
| LymphographyDS  | 0.232495 | 0.164399 | 0.232495 | 0.164399 | 0.284747 | 0.164399 | 0.164399 | 0.519875 | 0.164399 | 0.232495 |
| tic-tac-toeDS    | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.250522 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| waveformDS       | 0.628896 | 0.628896 | 0.618385 | 0.633719 | 0.625460 | 0.622896 | 0.628649 | 0.695701 | 0.627375 | 0.663450 |
| clean1dataDS     | 0.129461 | 0.204980 | 0.0000 | 0.0000 | 0.129461 | 0.158777 | 0.158777 | 0.183340 | 0.366679 | 0.091670 |
| SPECTDS         | 0.323230 | 0.323230 | 0.323230 | 0.345547 | 0.345547 | 0.323230 | 0.345547 | 0.423207 | 0.366508 | 0.345547 |
| ZooDS           | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.2000 | 0.0000 | 0.0000 |
| ecoliDS         | 1.052209 | 1.052209 | 1.052209 | 1.052209 | 1.128632 | 1.052209 | 1.354006 | 1.930334 | 1.052209 | 1.463850 |
| CongressEWDS     | 0.095783 | 0.095783 | 0.095783 | 0.135457 | 0.135457 | 0.135457 | 0.095783 | 0.317675 | 0.095783 | 0.135457 |
| ExactlyDS        | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.609918 | 0.0000 | 0.0000 |
| Exactly2DS       | 0.477493 | 0.456070 | 0.469042 | 0.481664 | 0.464758 | 0.481664 | 0.451664 | 0.536656 | 0.456070 | 0.447214 |
| M-of-nDS         | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.438178 | 0.0000 | 0.0000 |
| VoteDS          | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.230940 | 0.0000 | 0.115470 |
| kruskpDS         | 0.127555 | 0.117334 | 0.132370 | 0.122551 | 0.122551 | 0.122551 | 0.132370 | 0.229272 | 0.117334 | 0.150094 |
| heartDS          | 0.244339 | 0.244339 | 0.244339 | 0.211604 | 0.299253 | 0.299253 | 0.273179 | 0.405190 | 0.244339 | 0.366508 |

The maximum values of each competitor algorithm in each benchmark dataset are reported as in Table 14. From the Table, the WHOW algorithm achieves the best values in just 2 of the benchmark datasets.

The maximum values of each competitor algorithm in each benchmark dataset are reported as in Table 14. From the Table, the WHOW algorithm achieved the best values in 15 datasets, whereas the MPA reached the second rank by achieving the optimal values in 12 datasets. The PSO showed better results in six datasets and ranked third among the competitors, while the SCA, MVO, and WOA algorithms ranked fourth, achieving the best values in 4 benchmarks. The SMA algorithm ranked the least as it did not show optimal results in all datasets.

Table 15 shows the accuracy results for each competitor in the datasets. The Table shows that the WHOW algorithm achieved the highest accuracy in 17 benchmarks. The MPA algorithm ranked second and performed well in 7 datasets. The GA and SCA ranked third, with both PSO and MVO ranking fourth. The SMA algorithms ranked the least in accuracy in all benchmark datasets.
The WOA algorithm ranked first with the least computational time in ten benchmark datasets. The SMA ranked second and converged quickly in 6 datasets. Similarly, WHOW converges well in 3 benchmark datasets. The SCA algorithm converges quickly in only one dataset, whereas the rest of the competitor algorithms require a longer computational time in all tested datasets.

Table 17 presents the number of features selected using each competitor’s algorithm method. The Table showed that the WHOW selected the lowest features in 13 datasets, followed by the SCA with the lowest features in 6 benchmarks. The WHO, PSO, SSA, and MPA selected the lowest features in only one benchmark. The GA, MVO, SMA, and WOA failed to select the least number of features in all 20 benchmark datasets. Moreover, Figure 3 shows the average selected features for all datasets.

Table 18 presents the results of the statistical test of Wilcoxon’s rank-sum (i.e., \( P \)-value) used to examine the given the computational time shown in Table 16, the WOA algorithm ranked first with the least computational time in ten benchmark datasets. The SMA ranked second and converged quickly in 6 datasets. Similarly, WHOW converges well in 3 benchmark datasets. The SCA algorithm converges quickly in only one dataset, whereas the rest of the competitor algorithms require a longer computational time in all tested datasets.

Table 17 presents the number of features selected using each competitor’s algorithm method. The Table showed that the WHOW selected the lowest features in 13 datasets, followed by the SCA with the lowest features in 6 benchmarks. The WHO, PSO, SSA, and MPA selected the lowest features in only one benchmark. The GA, MVO, SMA, and WOA failed to select the least number of features in all 20 benchmark datasets. Moreover, Figure 3 shows the average selected features for all datasets.

Table 18 presents the results of the statistical test of Wilcoxon’s rank-sum (i.e., \( P \)-value) used to examine the given the computational time shown in Table 16, the WOA algorithm ranked first with the least computational time in ten benchmark datasets. The SMA ranked second and converged quickly in 6 datasets. Similarly, WHOW converges well in 3 benchmark datasets. The SCA algorithm converges quickly in only one dataset, whereas the rest of the competitor algorithms require a longer computational time in all tested datasets.

Table 17 presents the number of features selected using each competitor’s algorithm method. The Table showed that the WHOW selected the lowest features in 13 datasets, followed by the SCA with the lowest features in 6 benchmarks. The WHO, PSO, SSA, and MPA selected the lowest features in only one benchmark. The GA, MVO, SMA, and WOA failed to select the least number of features in all 20 benchmark datasets. Moreover, Figure 3 shows the average selected features for all datasets.

Table 18 presents the results of the statistical test of Wilcoxon’s rank-sum (i.e., \( P \)-value) used to examine the
TABLE 17. Selected features by each method.

| STD         | WHOW | WHO | PSO | GA | SCA | MVO | SSA | SMA | MPA | WOA |
|-------------|------|-----|-----|----|-----|-----|-----|-----|-----|-----|
| breastWDBCDS | 0.3978 | 0.5591 | 0.5161 | 0.5220 | 0.3456 | 0.5161 | 0.5279 | 0.6022 | 0.3871 | 0.6194 |
| ionosphereDS | 0.3088 | 0.4941 | 0.4588 | 0.4529 | 0.4044 | 0.4588 | 0.4765 | 0.3824 | 0.3353 | 0.4353 |
| wineDS | 0.4615 | 0.5077 | 0.5385 | 0.5538 | 0.4615 | 0.5385 | 0.5077 | 0.4769 | 0.5846 | 0.6154 |
| breastcancerDS | 0.2353 | 0.2941 | 0.4706 | 0.4588 | 0.5196 | 0.4471 | 0.4471 | 0.4000 | 0.3176 | 0.5176 |
| glassDS | 0.5000 | 0.6222 | 0.6667 | 0.6222 | 0.5556 | 0.6889 | 0.5333 | 0.5556 | 0.5926 | 0.6444 |
| sonarDS | 0.4116 | 0.4867 | 0.4967 | 0.4867 | 0.4116 | 0.5067 | 0.4733 | 0.5333 | 0.4233 | 0.4611 |
| LymphographyDS | 0.5111 | 0.5111 | 0.5889 | 0.5444 | 0.5556 | 0.5667 | 0.5778 | 0.5333 | 0.5222 | 0.6000 |
| tic-tac-toeDS | 0.6923 | 0.9556 | 1.0000 | 0.9778 | 0.7222 | 0.9831 | 0.9111 | 0.7222 | 0.9815 | 1.0000 |
| waveformDS | 0.6726 | 0.7333 | 0.7333 | 0.6952 | 0.6984 | 0.7302 | 0.6952 | 0.7143 | 0.7222 | 0.8476 |
| cleanlidataDS | 0.4598 | 0.4892 | 0.4964 | 0.4880 | 0.4955 | 0.5084 | 0.4735 | 0.6606 | 0.4639 | 0.4880 |
| SPECTDS | 0.4470 | 0.4455 | 0.3909 | 0.5364 | 0.2955 | 0.4818 | 0.4636 | 0.4909 | 0.4470 | 0.3818 |
| ZooDS | 0.4844 | 0.5875 | 0.5375 | 0.6000 | 0.4844 | 0.5875 | 0.5875 | 0.5833 | 0.5500 | 0.5875 |
| ecoliDS | 0.8750 | 0.7500 | 0.8571 | 0.8857 | 0.6667 | 0.8286 | 0.5143 | 0.5714 | 0.8000 | 0.6857 |
| CongressEWDS | 0.4375 | 0.4750 | 0.4750 | 0.5125 | 0.5000 | 0.4125 | 0.4750 | 0.4125 | 0.3875 | 0.4375 |
| ExactlyDS | 0.5128 | 0.5692 | 0.5385 | 0.5385 | 0.5385 | 0.5385 | 0.5385 | 0.5214 | 0.5231 | 0.5846 |
| Exactly2DS | 0.5641 | 0.7231 | 0.4000 | 0.6462 | 0.6154 | 0.5192 | 0.6642 | 0.5385 | 0.7500 | 0.5962 |
| M-of-nDS | 0.5000 | 0.5538 | 0.5231 | 0.5846 | 0.5077 | 0.5385 | 0.5846 | 0.6154 | 0.5538 | 0.5077 |
| VoteDS | 0.4583 | 0.3375 | 0.4750 | 0.4250 | 0.3333 | 0.4750 | 0.4500 | 0.4500 | 0.3625 | 0.4375 |
| kruskpDS | 0.5185 | 0.5944 | 0.5500 | 0.6278 | 0.3750 | 0.5500 | 0.5278 | 0.5278 | 0.6056 | 0.5722 |
| heartDS | 0.4712 | 0.6000 | 0.6000 | 0.6308 | 0.5231 | 0.5231 | 0.4769 | 0.5846 | 0.5538 | 0.5692 |

A significant difference between the WHOW and the other competitors in FS datasets. The Table shows there is a significant difference between the WHOW and the other algorithms in most datasets. In particular, it outperformed the GA in 35% of datasets and 40% of all datasets compared with MVO and WHO. Table 19 shows the results of the Friedman test. From the Table, WHOW outperformed the other algorithms and was ranked first in 18 of the 20 datasets. The MPA, PSO, GA, and MVO respectively follow the WHOW in order of the performance ranking.

Generally, the proposed WHOW method outperforms the other optimization algorithms in both experiments because the strategy of spiral-updating position helped the WHOW update its solutions within the optimization process. Therefore, it successfully escapes from local optima in most cases and effectively explores the search space. However, the proposed WHOW method has limitations, such as the computational time, where the WHOW could not outperform the other methods in more than 50% of all datasets. Therefore, this study recommends that the complexities of the proposed method be explored and improved in future studies.
VI. CONCLUSION AND FUTURE RESEARCH DIRECTIONS

In this study, we proposed a novel WHOW by integrating the strategy of the spiral movement of the WOA. Thus, it enhances the local exploitation ability of the WHOW, thereby ensuring intensification and diversification balancing. We evaluated the accuracy of the WHOW algorithm using CEC2019 test functions and compared the results with those of other popular swarm-based algorithms. The results showed that the WHOW algorithm provides a good ability in solving test function problems. Furthermore, we used the CEC2019 test functions to evaluate WHOW accuracy and compared the results with well-known and new algorithms. The results showed that the proposed WHOW algorithm also has an excellent ability to solve test function problems. Also, we used 20 more benchmark datasets from various fields to further examine the efficiency of the proposed WHOW algorithm for FS. The results showed that the proposed WHOW algorithm performed well in the selection of the most prominent subset of features. Thus, the simplicity of WHOW and the ability to deliver efficient and effective results showed that the proposed WHOW algorithm is an excellent improved version of the ordinary WHO with an enhanced exploration behavior. Future work should evaluate the proposed method in multi-objective optimization and parameter estimation. It could also be tested in image segmentation and solving multi-objective optimization and parameter estimation.

ACKNOWLEDGMENT

We sincerely acknowledge the support from Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R138), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

REFERENCES

[1] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*, vol. 454, Springer, 2012.

[2] M. Mafarja, A. Qasem, A. A. Heidari, I. Aljarah, H. Faris, and S. Mirjalili, "Efficient hybrid nature-inspired binary optimizers for feature selection," *Cognit. Comput.*, vol. 12, no. 1, pp. 150–175, Jan. 2020.

[3] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.

[4] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, nos. 1–2, pp. 273–324, 1997.

[5] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[6] I. Ahmadianfar, O. Bozorg-Haddad, and X. Chu, "Gradient-based optimizer: A new metaheuristic optimization algorithm," *Inf. Sci.*, vol. 540, pp. 131–159, Nov. 2020.

[7] T. J. Ypma, "Historical development of the Newton–Raphson method," *SIAM Rev.*, vol. 37, no. 4, pp. 531–551, 1995.

[8] J. J. Moré, "‘The Levenberg–Marquardt algorithm: Implementation and theory,‘ in *Numerical Analysis* (Lecture Notes in Mathematics), vol. 630, G. A. Watson, Ed. Berlin, Germany: Springer, 1978, doi: 10.1007/BFb0067700.

[9] S. M. A. Pahnehkolaei, A. Alfi, A. Sadollah, and J. H. Kim, "Gradient-based water cycle algorithm with evaporation and applied to chaos suppression," *Expert Syst. Appl.*, vol. 145, May 2020, Art. no. 113103.

[10] S. Deb, D. S. Abdelnur, M. Said, and E. H. Houssein, "Recent methodology-based gradient-based optimizer for economic load dispatch problem," *IEEE Access*, vol. 9, pp. 44322–44338, 2021.

[11] A. Mirjalili, F. Gandomi, S. Mirjalili, S. Saremi, H. Faris, and S. Mirjalili, "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," *Adv. Eng. Softw.*, vol. 135, Sep. 2019, Art. no. 102685.

[12] A. T. Azar, N. Banu, and A. Koubaa, "Rough set based ant-lion optimizer for feature selection," in *Proc. 6th Conf. Data Sci. Mach. Learn. Appl. (DSMA)*, Mar. 2020, pp. 50–54.

[13] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. Mirjalili, "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," *Adv. Eng. Softw.*, vol. 114, pp. 163–191, Dec. 2017.

[14] N. Nogaz, A. A. Ewees, M. A. Elaziz, and M. Mafarja, "Boosting salp swarm algorithm by sine cosine algorithm and disrupt operator for feature selection," *Expert Syst. Appl.*, vol. 145, May 2020, Art. no. 113103.
D. Rodrigues, X. S. Yang, A. N. de Souza, and J. P. Papa, “Binary flower optimization algorithm with crossover operators for feature selection and solving engineering problems,” IEEE Access, vol. 10, pp. 23304–23320, 2022.

M. H. Nadimi-Shahraki, M. Banaie-Dezfouli, H. Zamanii, S. Taghian, and S. Mirjalili, “B-MFO: A binary moth-flame optimization for feature selection from medical datasets,” Computers, vol. 10, no. 11, p. 136, Oct. 2021.

M. A. Elaziz, A. A. Ewees, R. A. Ibrahim, and S. Lu, “Opposition-based moth-flame optimization improved by differential evolution for feature selection,” Math. Comput. Simul., vol. 168, pp. 48–75, Feb. 2020.

A. H. Kashan, “League championship algorithm (LCA): An algorithm for global optimization inspired by sport championships,” Appl. Soft Comput., vol. 16, pp. 171–200, Mar. 2014.

H. Bingol and B. Atalas, “Chaotic league championship algorithms,” Aribun J. Sci. Eng., vol. 41, no. 12, pp. 5123–5147, Dec. 2016.

A. H. Kashan, “An effective algorithm for constrained optimization based on optics inspired optimization (OIO),” Comput.-Aided Des., vol. 63, pp. 52–71, Jun. 2015.

H. Bingol and B. Atalas, “Chaos based optics inspired optimization algorithms as global solution search approach,” Chaos, Solitons Fractals, vol. 141, Dec. 2020, Art. no. 110164.

B. Atalas and H. Bingol, “A physics based novel approach for travelling tournament problem: Optics inspired optimization,” Inf. Technol. Control, vol. 48, pp. 373–388, Sep. 2019.

B. Atalas and H. Bingol, “Comparative assessment of light-based intelligent search and optimization algorithms,” Light Eng., vol. 28, no. 6, pp. 51–59, 2020.

A. E. Ezugwu, O. J. Adeleke, A. A. Akinyelu, and S. Viriri, “A conceptual comparison of several Metaheuristic algorithms on continuous optimisation problems,” Neural Comput. Appl., vol. 32, no. 10, pp. 6207–6251, May 2020.

A. Ramadan, S. Kamel, I. B. M. Taha, and M. Tostado-Véliz, “Parameter estimation of modified double-diode and triple-diode photovoltaic models based on wild horse optimizer,” Electronics, vol. 10, no. 18, p. 2308, Sep. 2021.

M. H. Ali, S. Kamel, M. H. Hassan, M. Tostado-Véliz, and H. M. Zawbaa, “An improved wild horse optimization algorithm for reliability based optimal DG planning of radial distribution networks,” Energy Rep., vol. 8, pp. 582–604, Nov. 2022.

A. H. Gandomi, X.-S. Yang, and A. H. Alavi, “Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems,” Eng. Comput., vol. 29, no. 1, pp. 17–35, Jan. 2013.

A. A. Al-Saati, “The exploration of wild horse optimization in reliability redundancy allocation problems,” Intell. Eng. Syst. Comput., vol. 15, no. 4, p. 198, 2022.

R. Zheng, A. G. Hussian, H.-M. Jia, L. Abualigah, S. Wang, and D. Wu, “An improved wild horse optimizer for solving optimization problems,” Mathematics, vol. 10, no. 8, p. 1311, Apr. 2022.

M. Tubishat, M. A. M. Abushariah, N. Idris, and I. Alajjar, “Improved whale optimization algorithm for feature selection in Arabic sentiment analysis,” Int. J. Speech Technol., vol. 49, no. 5, pp. 1688–1707, May 2019.

D. Karaboga and S. Oktay, “A simple and global optimization algorithm for engineering problems: Differential evolution algorithm,” Turkish J. Elect. Eng. Comput. Sci., vol. 12, no. 1, pp. 53–60, 2004.

R. K. Agrawal, B. Kaur, and S. Sharma, “Quantum based whale optimization algorithm for wrapper feature selection,” Appl. Soft Comput., vol. 89, Apr. 2020, Art. no. 106902.

A. G. Hussian, A. E. Hassanieh, E. H. Houssein, S. Bhattacharyya, and M. Amin, “S-shaped binary whale optimization algorithm for feature selection,” in Recent Trends in Signal and Image Processing (Advances in Intelligent Systems and Computing), vol. 727, Singapore: Springer, 2019, pp. 79–87, doi: 10.1007/978-981-10-8863-6_9.

M. Mafarja, I. Jaber, S. Ahmed, and T. Chantar, “A novel enhanced evolutionary Student performance prediction model using whale optimization algorithm boosted with sine-cosine mechanism,” Appl. Sci., vol. 11, no. 21, p. 10237, Nov. 2021.

I. N. A. AL-Saati, “The exploration of wild horse optimization in reliability redundancy allocation problems,” IEEE Access, vol. 7, pp. 65359–65374, Apr. 2019.

S. Mirjalili and A. Lewis, “The whale optimization algorithm,” Adv. Eng. Softw., vol. 95, pp. 51–67, May 2016.

J. Kennedy and R. Eberhart, “Particle swarm optimization,” in Proc. IEEE ICWN, vol. 4, Nov./Dec. 1995, pp. 1942–1948.

M. Mitchell, An Introduction to Genetic Algorithms. Cambridge, MA: MIT Press, 1998.

S. Mirjalili, S. M. Mirjalili, and A. H. Gandomi, “Marine predators algorithm: A nature-inspired Metaheuristic,” Expert Syst. Appl., vol. 152, Aug. 2020, Art. no. 113377.

K. V. Price, N. H. Awad, M. Z. A. Ali, and P. N. Sanganthan, “Problem definitions and evaluation criteria for the 100-digit challenge special session and competition on single objective numerical optimization,” Nanyang Techn. Univ., Singapore: Tech. Rep., 2018.

D. Dua and C. Graff, “UCI machine learning repository,” School Inf. Comput. Sci., Univ. California, Irvine, Irvine, CA, USA, 2017. [Online]. Available: http://archive.ics.uci.edu/ml

AHMED A. EWEEES (Senior Member, IEEE) received the Ph.D. degree from Damietta University, Egypt. He is currently an Associate Professor of computer science with Damietta University. He co-supervises the master’s and Ph.D. students and leads and supervises various graduation projects. He has published over 100 scientific research papers in international journals and conferences. His research interests include machine learning, artificial intelligence, text mining, natural image processing, and metaheuristic optimization techniques.

FATMA H. ISMAIL received the master’s and Ph.D. degrees from the Faculty of Engineering, Ain Shams University, where she received the graduate degree. She worked in database and software industries for eight years. She is an Associate Professor of computer science with Misr International University. Her main research interests include data mining, machine learning, computational intelligence and swarm optimization, and deep learning.

RANIA M. GHIOMI received the Ph.D. degree in integrating artificial intelligence techniques in medical diagnosis from Mansoura University, Egypt. She is with the Department of Information Technology, College of Computer and Information Sciences, Princess Nourah Bint Abdullah University, Riyadh, Saudi Arabia. Her research interests include computer-aided diagnostic systems, medical image and sound processing, natural language processing, deep learning, and meta-heuristic optimization methods.

A. A. Ewees et al.: Wild Horse Optimizer-Based Spiral Updating for Feature Selection