Efficient Hybrid Nature-Inspired Binary Optimizers for Feature Selection

Article *in* Cognitive Computation · July 2019

DOI: 10.1007/s12559-019-09668-6

6 authors, including:

Majdi Mafarja
Birzeit University
49 PUBLICATIONS **956** CITATIONS

Ali Asghar Heidari
National University of Singapore
42 PUBLICATIONS **838** CITATIONS

Ibrahim Aljarah
University of Jordan
100 PUBLICATIONS **1,569** CITATIONS

Hossam Faris
University of Jordan
149 PUBLICATIONS **2,189** CITATIONS

Cognitive Computation
https://doi.org/10.1007/s12559-019-09668-6

Efficient Hybrid Nature-Inspired Binary Optimizers for Feature Selection

Majdi Mafarja[1] · Asma Qasem[2] · Ali Asghar Heidari[3,4] · Ibrahim Aljarah[5] · Hossam Faris[5] · Seyedali Mirjalili[6]

Received: 14 November 2018 / Accepted: 3 July 2019

Abstract

The process of dimensionality reduction is a crucial solution to deal with the dimensionality problem that may be faced when dealing with the majority of machine learning techniques. This paper proposes an enhanced hybrid metaheuristic approach using grey wolf optimizer (GWO) and whale optimization algorithm (WOA) to develop a wrapper-based feature selection method. The main objective of the proposed technique is to alleviate the drawbacks of both algorithms, including immature convergence and stagnation to local optima (LO). The hybridization is done with improvements in the mechanisms of both algorithms. To confirm the stability of the proposed approach, 18 well-known datasets are employed from the UCI repository. Furthermore, the classification accuracy, number of selected features, fitness values, and run time matrices are collected and compared with a set of well-known feature selection approaches in the literature. The results show the superiority of the proposed approach compared with both GWO and WOA. The results also show that the proposed hybrid technique outperforms other state-of-the-art approaches, significantly.

Keywords Whale optimization algorithm · Grey wolf optimizer · Optimization · Feature selection · Metaheuristics

Introduction

The continuous rapid growth in the amount of data stored, processed, or retrieved from information systems has made the process of extracting useful information [16] more difficult than before. Mainly, collected data commonly consist of redundant and irrelevant data that jeopardize the success and performance of learning algorithms (e.g., classification). Thus, a tool to eliminate those redundant features and minimize their adverse impacts is essential. Dimensionality reduction (DR) is a pre-processing technique that is responsible for eliminating the redundant and irrelevant data from the dataset being processed [29].

Generally speaking, DR techniques can be categorized into two types: feature extraction (FE) and feature selection (FS). In the former class, the aim to project the original high-dimensional data into a new feature space of low dimensionality is through extracting new features from the existing features in the dataset. In the latter class, however, FS methods tend to select the least number of features that can effectively represent the original dataset. FS is widely used in different fields such as pattern recognition, text mining, and signal processing, to name a few [31].

FS contributes to enhancing generalization by reducing over-fitting, computation time, and storage, enhancing the classification accuracy by creating an accurate predictive model among other things [1, 49]. FS process consists of four main stages as follows [5]:

- Feature subset generation: a search procedure that selects a feature subset through one of the following search strategies: complete search, random search, or heuristic search.
- Feature subset evaluation: this item is used to evaluate the generated subsets using filters, wrappers, and embedded models.
- Stopping criterion: this item determines when the optimization process should stop. Examples of stopping criteria are the maximum number of iterations or setting a target goal like an absolute classification accuracy.
- Validation: this item validates the accuracy of results that are produced when using the generated subsets in an optimization algorithm.

✉ Ibrahim Aljarah
i.aljarah@ju.edu.jo
http://www.evo-ml.com

Extended author information available on the last page of the article.

Published online: 29 July 2019

The most important aspects of any FS method are how to search for a feature subset and evaluate the selected subset. When considering the search method, three main strategies can be investigated: complete, random, and heuristic search algorithms. In a complete search, as the name indicates, all possible combinations of feature subsets should be considered. It is evident that the complete search ensures finding the optimal subset since it investigates all subsets in the search space. However, the complexity of feature selection is exponential (2^N, where N is the number of features in the dataset), so exact methods suffer from slow speed and are impractical for large-scale datasets.

As an alternative strategy, a random search can be used. The best case from the random search strategy is that it is unnecessary to wait until the search is completed since the optimal subset can be found during the early stages. On the other hand, a random search will be equivalent to a complete search in the worst case in terms of complexity [36].

As the complete search is unfeasible with the high-dimensional datasets due to the high computation time required to find the optimal solution and the random search may behave like the complete search in the worst case, the third strategy, i.e., the heuristic search can be used. In a heuristic search, heuristic information is used to make "education decisions" and guide the search process. Such heuristic information is highly dependent on the problem and often cannot be generalized.

Metaheuristic algorithms are general-purpose search techniques that are mostly problem independent [45, 60]. The literature shows their efficiency and effectiveness in solving optimization problems compared with other approaches like complete and random search [58]. Examples of metaheuristic methods are genetic algorithm (GA) [26], particle swarm optimization (PSO) [7], ant colony optimization (ACO) [6], and differential evolution (DE) [54]. Some of the newest methods are whale optimization algorithm (WOA) [42], Harris Hawks Optimizer (HHO) [25], and grey wolf optimizer (GWO) [43].

FS methods can be categorized into three types by considering the evaluation criteria: filters, wrappers, and embedded methods. In the wrapper methods, the performance of a specific learning algorithm (e.g., classification) is used to define the goodness of a feature subset. In filter methods, however, the distance and information measures are usually used as evaluation criteria. Embedded FS methods are internally involved during the learning process (i.e. C4.5 decision tree classifier) [23]. As one of the major concern in the field of machine learning is to improve the predictive accuracy of an algorithm (e.g., classifier) and to achieve high learning performance from data, the wrapper model can be used [31].

This work is the first attempt to systematically develop a parallel hybrid of the GWO and WOA algorithms. Since feature selection problems have binary variables, the hybrid method is designed as a discrete optimizer for the first time in the literature as well. The rest of the paper is organized as follows:

The "Preliminaries" section provides preliminaries, essential definitions, and mathematical models of both GWO and WOA. The "The Proposed Approaches" sectionhybridizes both algorithms and proposes a binary optimization algorithm. The experimental results are presented and discussed in the"Experimental Results" section. Finally, "Conclusion and Future Directions" concludes the work and suggests future directions.

State-of-theArt

Many wrapper-based FS methods that use metaheuristic cores can be found in the literature. As the first example, PSO has been widely used within the FS methods. In [35, 37], for instance, two different FS techniques were proposed, where the performance of the PSO algorithm was studied with different updating strategies of the inertial weight parameter.

A binary version of the GWO (called BGWO) was proposed in [9]. Two methods were used to implement BGWO. In the first one, individual steps taken towards the best three solutions were changed to binary. Then, a crossover operator was employed to update the solutions. In the second approach, a sigmoidal function was applied to squash the continuously updated position then randomly threshold these values to find corresponding binary values. When compared with other works in the literature, BGWO was able to show a good performance. In [40], the authors proposed dragonfly algorithm (DA) that was used many times to solve FS problems [38] as well. In [34], the authors also evaluated the performance of binary DA for a series of FS cases. Another binary DA was used with time-varying transfer functions and the results show the significant impact of the used functions on the quality of features [32].

In [48], the authors presented a wrapper FS method using the GA algorithm as a search strategy with neural network classifier as an evaluation core. Recently, the well-known GA was also used as the searching core within a wrapper-based method for spam detection [14].

Emary et al. in [8] presented a binary version of the ant lion optimization (ALO) in the wrapper model to find the optimal subset and increase the classification accuracy. In [61], another wrapper FS method based on ALO was presented. More recently, two wrapper FS approaches, which employed a salp swarm algorithm (SSA) as a search strategy, were proposed in [2, 15]. An asynchronous SSA with more than one leader and accelerated convergence was utilized to solve FS problems and results are promising [2].

From the previously mentioned approaches, it can be seen that metaheuristic-based FS approaches showed good performance in selecting the most informative features that enhance the performance of classification algorithms [13, 56]. This fact is due to the ability of such techniques to find reasonably good solutions in a short period [55]. They do not search the entire search space but make “education decisions” to avoid searching unpromising regions of the search space.

Solutions to many theoretical and practical problems can be found using a proper mathematical modeling [17–21]. However, when designing a biology-inspired metaheuristic, two contradictory characteristics must be considered: exploration and exploitation, and it is important to find a balance between them [11, 24, 51]. High exploration reduces the quality of results, and fast convergence prevents the algorithm from finding a globally optimum solution [3, 22]. Nonetheless, high exploitation increases the chance of algorithm to stuck at LO and miss the best global solution, too. One way to achieve a convincing balance between exploitation and exploration is to use a hybrid model [4, 12]. In a hybrid model, two or more metaheuristic algorithms are used to develop a new algorithm with higher performance rather than using each algorithm alone.

In the field of optimization, the interest in using hybrid biology-inspired algorithms has risen considerably over the last years. Using hybrid models provided the best results for many optimization problems [55]. This is the case with FS problems too as the literature shows. For instance, a hybrid WOA with simulated annealing (SA) algorithm was proposed in [39], in which two hybrid models were employed: low-level teamwork hybrid and high level rely on hybrid. In the first approach, the SA was embedded in the WOA to enhance its exploitation ability. In the second model, the SA was employed after WOA terminates to improve the global best solution. As the experimental results showed in [39], the proposed approach outlined a good performance when compared with other similar methods in the literature. In [62], a hybrid FS method that combines the DF and ABC algorithms was proposed to alleviate their drawbacks. The proposed hybrid method provides a solution to the general classification task in data mining. The fast convergence nature of the DF algorithm was merged with the new binary mutation phase. The proposed approach modifies the process of onlooker bee to prevent it from stagnation in LO.

Another hybrid biology-inspired algorithm that combined the idea of evolutionary algorithms with swarm-based algorithms was proposed in [33]. In this approach, an evolutionary population dynamics (EPD) idea with a selection operator, was combined with grasshopper optimization algorithm (GOA). In [46], the authors proposed a hybrid FS method, where a local search algorithm was embedded in PSO to guide the search process to select the most useful features regarding their correlation information. When compared with five biology-inspired methods, the results proved the capability of the current approach to enhance classification accuracy.

Another modification to the PSO algorithm was presented in [59], which enhances its updating techniques of *pbest* and *gbest*. In addition, Nguyen et al. [47] developed a hybrid novel method that depends on PSO for FS. A local search technique was developed to mimic the backward FS method. The algorithm uses a classification error rate as a wrapper-based fitness method and uses filter-based measure, in order to combine the advantages of wrapper and filter models. It was shown that this approach is superior when compared with three PSO-based optimizers and two traditional methods.

Despite the success of the abovementioned works, the no free lunch (NFL) theorem in optimization [57] has logically proved that there is no optimization algorithm for all optimization problems. In the area of feature selection, none of the above algorithms is able to find the optimal set of features for all datasets. This motivated our attempt to hybridized and improved two of the most popular recent algorithms GWO and WOA to solve feature selection problems.

There are three similar hybrids in the literature. In 2017, Mohamed et al. [44] proposed a sequential hybrid of WOA and GWO. Both algorithms were equipped with crossover and mutation operators too. They were employed to solve optimal power dispatch problems and showed that this model is superior as compared with both GWO and WOA individually. However, the main issue with this hybrid is its computational cost. The sequential hybridization of both algorithms require running the WOA first and using the best solution as the initial solution for GWO.

In 2018, Singh and Hachimi [53] integrated the spiral equation of WOA into GWO and apply it to structural design problems. A similar method proposed in [27] and applied to data clustering algorithms. Despite the merits of this hybrid, the authors used it for continuous optimization; hence, it is not able to deal with problems with binary variables.

Preliminaries

Grey Wolf Optimizer

GWO was proposed by Mirjalili in [43], simulating the hunting behavior of gray wolves and their leadership hierarchy. In the following paragraphs, the mathematical model of GWO as well as its main phases (social hierarchy, encircling prey, hunting, attacking, and searching for the prey) are explained.

Social Hierarchy

As mentioned in [43], there are four types of gray wolves: alpha, beta, delta, and omega. To mathematically model the hierarchy of wolves in nature, [43] assumed the first three best solutions are alpha (α), beta (β), delta (δ), and the rest are considered omega (ω). It is worth mentioning that GWO is guided by the α, β, and δ.

Encircling Prey

Gray wolves encircle the prey during the hunting process. Equation 1 is utilized to mathematically model the distances between each search agent (wolf) and the prey.

$$\vec{D} = |\vec{C} \times \vec{X}_P(t) - \vec{X}(t)| \tag{1}$$

where $\vec{X}_P(t)$ and $\vec{X}(t)$ represent the position vector of the prey and a wolf, respectively, and t is iteration, $\vec{C}$ is a coefficient vector, which is calculated as in Eq. 2, $\vec{r_2}$ represents a random vector in the interval [0, 1].

$$\vec{C} = 2 \times \vec{r_2} \tag{2}$$

To identify the next location of the prey, GWO have to apply Eq. 3:

$$\vec{X}(t+1) = \vec{X}_P(t) - \vec{A} \times \vec{D} \tag{3}$$

where $\vec{A}$ represents a coefficient vector, which is calculated as in Eq. 4, the elements of $\vec{a}$ are linearly decreased from 2 to 0 over the course of iterations, and r_1 is a random vector in [0, 1].

$$\vec{A} = 2\vec{a} \times \vec{r_1} - \vec{a} \tag{4}$$

According to Eqs. 1 and 3, a gray wolf can update its position based on the position of the prey. By adjusting the value of $\vec{A}$ and $\vec{C}$, different locations can be reached around the prey as well as, the random vectors r_1 and r_2 which enable search agents (wolves) to update their positions around prey in any random location. This assists GWO in accurately exploring the area around the prey.

Hunting phase

The hunting process is guided by the best three solutions obtained so far α, β, and δ [50, 52]. To mathematically model it, α, β, and δ solutions must identify the potential location of the prey. Each search agent in the population updates its position around the prey randomly as in Eq. 11. The following updating rules illustrate how wolves update their positions.

$$\vec{D}_\alpha = |\vec{C}_1 \times \vec{X}_\alpha - \vec{X}| \tag{5}$$

$$\vec{D}_\beta = |\vec{C}_2 \times \vec{X}_\beta - \vec{X}| \tag{6}$$

$$\vec{D}_\delta = |\vec{C}_3 \times \vec{X}_\delta - \vec{X}| \tag{7}$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \times (\vec{D}_\alpha), \tag{8}$$

$$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \times (\vec{D}_\beta) \tag{9}$$

$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \times (\vec{D}_\delta) \tag{10}$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{11}$$

where $\vec{X}_\alpha$, $\vec{X}_\beta$, and $\vec{X}_\delta$ are the first three best solutions in the population at a given iteration t, $\vec{D}_\alpha$, $\vec{D}_\beta$, and $\vec{D}_\delta$ are defined using Eqs. 5–7, $\vec{C}_1$, $\vec{C}_2$ and $\vec{C}_3$ are defined as in Eq. 15 and $\vec{A}_1$, $\vec{A}_2$, and $\vec{A}_3$ are defined as in Eq. 14.

Attacking the Prey (Exploitation)

To mathematically simulate the attacking to the prey, the value of $\vec{a}$ is linearly decreased from 2 to 0 over the course of iterations using Eq. 16. It is clear that the value of $\vec{A}$ decreases proportional to that of $\vec{a}$. When the value of $-1 < \vec{A} < 1$, the next location of the search agent takes place in any location between the prey and its current location.

Search for the Prey (Exploration)

Ssearching for the prey depends on the positions of alpha, beta, and delta and those wolves who move away from each others to expand their searching range. The mathematical model of this phase is based on using random values of vector $\vec{A}$. When $\left|\vec{A}\right| > 1$, the wolves shift away from the prey for the sake of finding a better prey (exploration).

Algorithm 1 presents the pseudo-code of GWO. It is worth mentioning that the updating of parameter a can assist in switching from exploration to exploitation trends.

Algorithm 1 Pseudocode of GWO.

Initiate the parameters (e.g., the maximum number of iterations L and population size)
Generate random population of wolves $X_i (i = 1, 2, \ldots, n)$
Initialize a, A, and C
Evaluate all wolves by calculating the fitness of all locations
Define X_α as the fittest search agent, X_β as the second best search agent, X_δ as the third best search agent
while $(t < L)$ **do**
  for each Wolf X_i in the population **do**
    Update the position of the current wolf by Eq. 11
  Update a, A, and C
  Evaluate all wolves in the population by calculating the fitness of each member
  Update X_α , X_β, and X_δ
  $t = t + 1$
return X_α

Whale Optimization Algorithm

WOA was proposed in [42] to simulate the hunting behavior of humpback whales. Humpback whales swim around the prey within shrinking circles and creating distinctive bubbles along a "9"-shaped path. In WOA, the exploitation phase is achieved by encircling the prey and spiral bubble net attacking method. For the exploration phase, the random search process for prey is performed. These phases are discussed in the following subsections.

Exploitation phase (Encircling Prey and Spiral Bubble Net Attacking Method)

As mentioned above, humpback whales begin their hunting process by encircling the prey. In order to model this behavior, Eqs. 12 and 13 is utilized [42].

$$D = | C.\mathbf{X}^*(t) - \mathbf{X}(t) | \tag{12}$$

$$\mathbf{X}(t+1) = \mathbf{X}^*(t) - \mathbf{A}.D \tag{13}$$

where t denotes the current iteration, X^* indicates the best solution found so far, X is the current whale, $|\ |$ is the absolute value and $<\cdot>$ is an element by element multiplication. Coefficient vectors A and C are calculated as in Eqs. 14 and 15.

$$\mathbf{A} = 2\mathbf{a}.\mathbf{r} - \mathbf{a} \tag{14}$$

$$\boldsymbol{C} = 2.\mathbf{r} \tag{15}$$

where $\overrightarrow{a}$ decreases linearly from 2 to 0 through the iterations and $\overrightarrow{r}$ is a random vector in [0,1]. From Eq. 13, the search agents (whales) update their positions based on the best-known solution position (prey). For controlling the exploitation of the feature space, WOA adjusts the values of A and C vectors.

The shrinking encircling behavior is performed by decreasing the value of a as in Eq. 14:

$$a = 2\left(1 - \frac{t}{L}\right) \tag{16}$$

where t is iteration and L is the maximum limit of iterations. To model the spiral-shaped path, the distance between the search agent and the best agent obtained so far is calculated, then, a spiral rule is used to calculate the position of the neighbor solutions as in Eq. 17:

$$\mathbf{X}(t+1) = D'.e^{bl}.cos(2\pi l) + \mathbf{X}^*(t) \tag{17}$$

$$D' = | C.\mathbf{X}^*(t) - \mathbf{X}(t) | \tag{18}$$

where D' represents the distance between the ith search agent and the best solution obtained so far, b is a constant and is used to define the shape of the logarithmic spiral, and l is a random number inside [− 1,1]. To select between the shrinking encircling and the spiral-shaped path, a probability of 50% is used as follows:

$$\mathbf{X}(t+1) = \begin{cases} \text{Shrinking Encircling by Eq.(13)} & p < 0.5 \\ \text{Spiral Shaped Path by Eq.(17)} & p \geq 0.5 \end{cases} \tag{19}$$

where p is a random number in [0,1].

Exploration Phase

As the main mechanism of exploration, the positions of agents are updated using the position of the current best solution. For further improvements on the exploration, a random agent is selected to change the positions away from the area of the best solution. This strategy can avoid WOA from being trapped in local solutions. To achieve this, the vector $\overrightarrow{A}$ with the random values less than − 1 or greater than 1 is used. The mathematical representation of this method represents as in Eqs. 20 and 21.

$$D = | C.\mathbf{X}_{\mathbf{rand}}(t) - \mathbf{X}(t) | \tag{20}$$

$$\mathbf{X}(t+1) = \mathbf{X}_{\mathbf{rand}}(t) - \mathbf{A}.D \tag{21}$$

where the $\mathbf{X_{rand}}$ is a search agent randomly selected from the current population.

The pseudo-code of WOA is shown in Algorithm 2.

Algorithm 2 Pseudo-code of WOA.

Initiate the parameters (e.g., the maximum number of iterations L and population size)
Generate random whales $X_i(i = 1, 2, \ldots, n)$
Calculate the fitness values
Define X^* as the best solution
while $(t < L)$ **do**
  for each whale **do**
    Update a, A, C, l, and p
    if $(p < 0.5)$ **then**
      if $(\mid A \mid < 1)$ **then**
        Update the current whale by Eq. 13
      else if $(\mid A \mid > 1)$ **then**
        Select a random solution from a population
        Update the position of the current solution by Eq. 21
    else if $(p > 0.5)$ **then**
      Update the position of the current solution by Eq. 17
  Check if any solution goes beyond the search space and adjust it
  Calculate the fitness of each solution
  Update X^* if there is a better solution
  $t = t + 1$
return X^*

WOA generates and the initial population of solutions and then evaluates all of them by calculating their fitness values. This allows finding the best solution too. In the next phase, the algorithm goes into three steps and repeats them until a stopping constraint. During the first step, the coefficients are updated. In the second step, WOA generates a random value and uses it to update the position of a solution by using one of the following Eqs. 13/21, or 17. In the third step, the algorithm checks if the solutions go beyond the search space and then adjust them. Finally, the algorithm returns the best solution.

The current best solution is used to update the position of the remaining solutions to guarantee the convergence of the algorithm. However, this may lead the algorithm towards a locally optimal solution. For this reason, the algorithm uses a random search agent from a current population to provide more chances for exploring the target search area, as well as the switching ability between Eqs. 13/21 or Eq. 17 to update the position of solutions. To balance exploitation and exploration, WOA uses adaptive values for parameter a. This parameter is responsible for smoothly reducing the adjustments in the solutions to make convergence proportionate to the iterations number.

Binary Optimization

In general, metaheuristics are originally developed to solve continuous optimization problems. However, many optimization problems have a binary search space (e.g. FS problems). An algorithm should be altered to solve binary problems. In this subsection, binary versions of GWO and WOA are introduced.

According to [41], one of the simplest methods to convert a continuous to a binary version is to use the transfer functions (TFs). Their simplicity comes from the fact the they do not change the structure of the original version of the continuous algorithm, except for minor changes to some operators. TFs are used to calculate the probability of mapping the continuous value in a solution to a binary value.

According to the nature of FS search spaces where a feature is either selected (1) or not selected (0), a binary version of the GWO and WOA should be used. In the following subsections, the binary versions of these algorithms are described.

Binary of WOA for FS

In order to convert the continuous WOA to the binary version, the sigmoidal transfer function (shown in Fig.1)

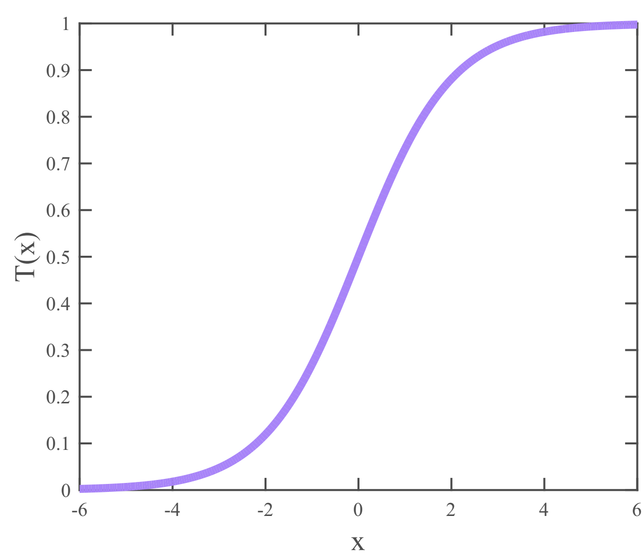

Fig. 1 Sigmoid transfer function

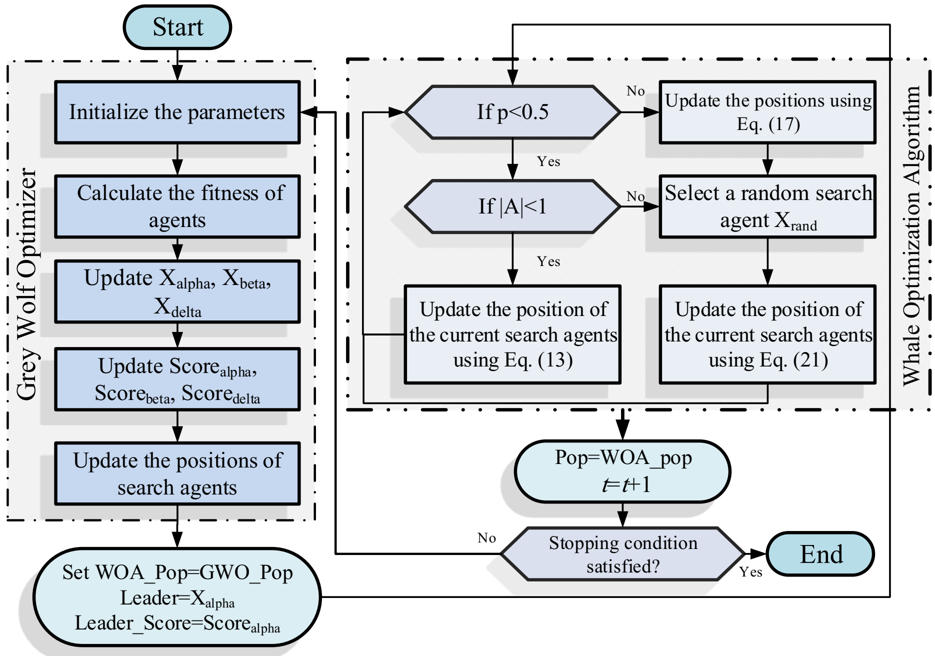

Fig. 2 Structure of HSGW approach

is used. This transfer function receives a solution in a continuous form and re-defines it as given in Eq. 22.

$$T(\triangle X_t) = \frac{1}{1+e^{-\triangle X_t}} \tag{22}$$

where $\triangle X_t$ indicates the step vector of the search space at iteration t. Now, the position of the current search agent (Whale) will be updated as given in Eq. 23.

$$X^d_{t+1}(t+1) = \begin{cases} 1 & If\, u < T(\triangle X_{t+1}) \\ 0 & If\, u \geq T(\triangle X_{t+1}) \end{cases} \tag{23}$$

where u is a random number with uniform distribution in (0, 1) and $X^d_{t+1}(t+1)$ indicates the binary step in dimension d.

Binary of GWO for FS

In the proposed approach, the same transfer function that was used with the WOA, is used to build the binary version of GWO. Each of best three solutions (i.e., X_1, X_2, and X_3) is passed to the sigmoid function (Eq. 22) as input, then the result represent a probability of converting an element of the continuous solution to 0 or 1 as described in Eq. 24.

$$x^d(t) = \begin{cases} 1 & \text{rand} < T(\text{Sol}(t)) \\ 0 & \text{rand} \geq T(\text{Sol}(t)) \end{cases} \tag{24}$$

where Sol is replaced by X_1, X_2, and X_3.

Finally, the binary version of a solution x is calculated as explained in Eq. 25:

$$x^d(t) = \begin{cases} 1 & x^d(t) + T(\text{Sol}(t)) \geq 1 \\ 0 & \text{otherwise} \end{cases} \tag{25}$$

The Proposed Approaches

GWO is a recent metaheuristic algorithm, which shows superiority in tackling a number of optimization problems. In GWO, each solution is updated according to the position of three best solutions in the population, which makes the algorithm more oriented to the exploitation than to the exploration. In contrast, WOA considers repositioning some

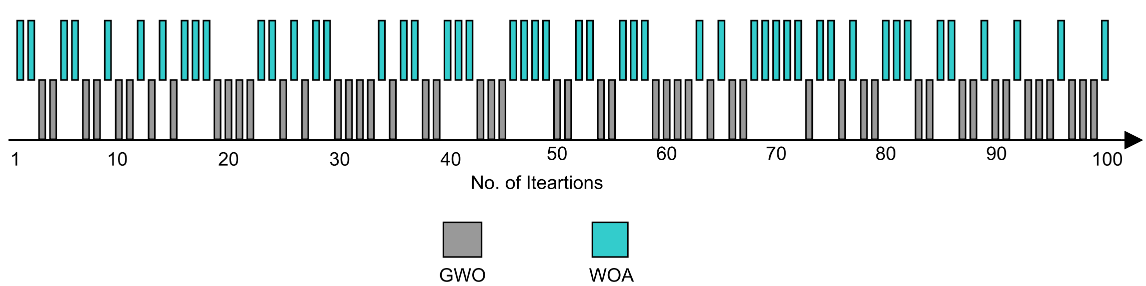

Fig. 3 Switching behavior of RSGW in a sample run. In this observation, the GWO is selected in 51 iterations, and WOA is selected in 49 iterations

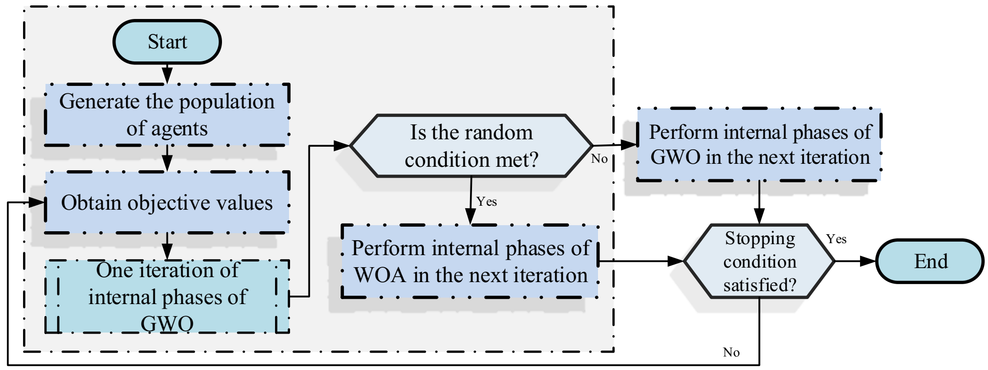

Fig. 4 Structure of RSGW approach

solutions around a randomly generated solution during the optimization process. This helps the algorithm to escape from locally optimal solutions. Hence, based on structural analysis, the exploration potential in WOA is higher than the GWO. For both algorithms, as for any metaheuristic, having a fine balance between exploration and exploitation has a significant impact on enhancing the performance of the algorithm. One of the possible ways to achieve this balance and avoid premature convergence is to efficiently utilize the advantages and operators of both algorithms, that each one has its own capabilities and merits.

The authors in [55] proposed a taxonomy for the possible hybrid strategies where four basic hybridization strategies can be derived:

- LRH (low-level relay hybrid): a metaheuristic algorithm is embedded into a single-solution algorithm.
- HRH (high-level relay hybrid): Two metaheuristics are executed in sequence in homogenous (same algorithms) or heterogeneous (different algorithms) manners.
- LTH (low-level teamwork hybrid): A metaheuristic algorithm is embedded into a population-based algorithm.
- HTH (High-level teamwork hybrid): Two metaheuristics are executed in parallel.

In this work, we propose a combination of two metaheuristics (GWO and WOA) following the high-level relay hybrid (HRH) group. As the hybrid approaches have different performance for the different hybridization strategies, in this work, we proposed three different strategies; serial grey-whale optimizers, random switching grey-whale optimizers, and adaptive switching grey-whale optimizers.

Hybrid Serial GWO-WOA

In this approach, a solution is updated in two phases in each iteration. In the first phase, a solution is updated by applying the operators of GWO, and it is re-positioned around the best three solutions in the population denoted by α, β, and δ.

In the second phase, the WOA operators are employed to extend the search process to other promising areas in the feature space. The key point in this hybrid model is that WOA works on the same population used in the GWO in the first phase, but the leader is α wolf, which should be updated after re-positioning the solutions manipulated in the first phase. In other words, after re-positioning the current solution based on GWO's operators, it is evaluated, and if it becomes fitter than the best solution (i.e., α),

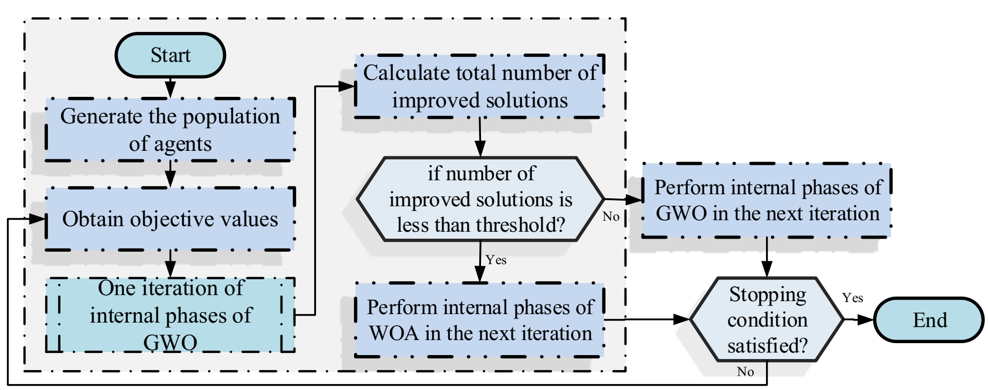

Fig. 5 Structure of ASGW approach

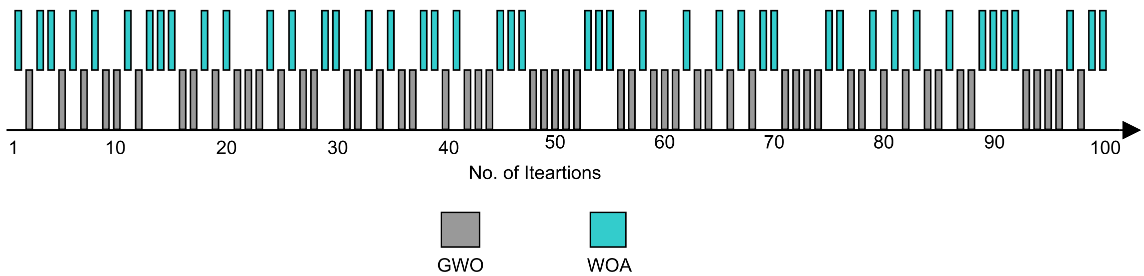

Fig. 6 Switching behavior of ASGW in a sample run. In this observation, the GWO is selected in 55 iterations, and WOA is selected in 45 iterations

then, α solution is updated and used as the leader for the WOA algorithm. These two phases are repeated until the stopping condition (i.e., max number of iterations) is met. The structure of hybrid serial GWO-WOA (HSGW) is shown in Fig. 2.

Random Switcher GWO-WOA

As the purpose of hybridizing these two algorithms is to help GWO to escape from LO, we proposed to use of one algorithm in each iteration instead of employing both GWO and WOA. It is obvious that using one algorithm requires shorter computational time to locate the (near) optimal solution. In this approach, the proposed procedure can randomly switch between two algorithms, where a random number in [0, 1] is generated if its value is less than 0.5, then GWO is employed, and WOA is employed when the value is greater than 0.5.

To illustrate the switching process, Fig. 3 shows how the proposed approach switches between two selected engines during the exploration and exploitation processes. The structure of random switcher GWO-WOA (RSGW) is also represented in Fig. 4.

Adaptive Switcher GWO-WOA

In the previous approach, a stochastic factor is used to control the selection of operators from the GWO or WOA without considering any information about the performance of each algorithm. In this approach, we adopted an adaptive mechanism to control the selection process. The optimization process starts with the GWO algorithm. Then, at each iteration, the total number of improved solutions in the population is calculated. If the number of improved solutions is less than a threshold, the WOA will be employed in the next iteration. The motivation behind this approach is that if GWO could not improve the population, it might be stuck at LO and better solutions cannot be found. At this point, employing the WOA moves the current population into a different area where some better solutions may be found. The same process is repeated when employing the WOA. The structure of adaptive switcher GWO-WOA (ASGW) is shown in Fig. 5. Figure 6 also shows how ASGW selected one of the core methods during optimization.

F_1	F_2	F_3	F_4	...	F_{N-2}	F_{N-1}	F_N
1	0	1	1	...	0	1	0

Fig. 7 An example of a binary solution in FS

Solution Representation

The solution representation of optimization problems is one of the most important issues that should be considered when designing an optimization algorithm. An example of FS solution that adopted in this work is shown in Fig. 7. The solution is represented as an N-sized binary vector where N is the total number of features in a dataset. Each element in this solution indicates two options; either select a feature (1) or not (0). Based on that, the number of all possible feature subsets equals 2^N, and with this number of subsets, the complete search is infeasible. Metaheuristic algorithms are considered a good alternative in this case.

Table 1 Parameter settings

Parameter	Value
Population size	20
Number of iteration	100
Dimension	Number of features
Number of runs for each technique	30
α in fitness function	0.99
β in fitness function	0.01
a in GWO and WOA	[2 0]
G_0 in GSA	100
α in GSA	20

Fitness Function

FS is associated with more than one objective function, so it is considered a multi-objective optimization problem. The evaluation of generated subset depends on two criteria: the number of selected features that should be minimized and classification accuracy that should be maximized. These criteria consists of two contradictory purposes that must be considered when developing FS algorithms [10]. The first objective is the selected features number that should be minimized, while in the second objective, the accuracy of a classifier should be maximized. Equation 26 aggregate both objectives simultaneously and convert the problem into a single-objective problem. The fitness value is calculated for each solution in the generated population over the course of iterations, and the subset with a least fitness value is treated as the best solution. The minimum fitness value is the sum of minimum classification error rate and minimum number of selected features.

$$\text{Fitness} = \eta \times \epsilon + \mu \frac{|\upsilon|}{|\Upsilon|} \tag{26}$$

where η and μ constants represent the importance of the classification accuracy and the subset length, respectively. Note that $\eta + \mu = 1$. The classifier's error rate is represented by ϵ. υ is the number of selected attributes in the subset being evaluated while Υ is the total number of attributes in the dataset.

Experimental Results

In this section, the results of the proposed approaches are presented. A set of comparisons are conducted to assess the performance of the proposed approaches. Firstly, the results of the hybrid bGWO-WOA approaches (i.e., HGW, HGW, ASGW) are compared with the basic GWO and WOA approaches. A comparative study is then conducted with the state-of-the-art methods. Finally, a comparison is done with some well-known results in the literature of feature selection.

Experimental Setup

The proposed approaches are implemented using Matlab, and all experiments in this section were conducted on a PC with Intel Core(TM) i5-5200U 2.2GHz CPU and 4.0GB RAM. The sensitivity analysis for the proposed method is conducted as well (see "Parameter Tuning section). The experiments include studying the effect of the population size (N) parameter and the number of iterations on the GWO and WOA. The other standard parameter settings for GWO, WOA, and other algorithms are recorded in Table 1, which are set based on the recommendations from the original papers of GWO and WOA. To make fair comparisons and to be able to obtain reliable statistical analysis, each algorithm was executed 30 independent times.

Three different state-of-the-art FS algorithms (i.e., BGOA, BGSA, GA, and PSO) were implemented, and their results were recorded and compared with the proposed approaches. The abovementioned fitness function is utilized in all experiments to compromise between the classification accuracy and the number of selected features where the classification accuracy is based on the KNN classifier (where $K = 5$ [8, 9, 33]) with the Euclidean distance metric. Each dataset is divided into two sets; training and testing, the training set contains 80% of the instances while the remaining instances are used as a testing set.

The performance of the proposed approaches is assessed by conducting experiments on 18 datasets drawn from the UCI data repository [30]. Table 2 describes the properties of the datasets in terms of number of features, number of instances, number of classes, and data area.

Table 2 List of used datasets in the experiment (NF, no. of features, NI, no. of instances, NC, no. of classes)

Dataset	NF	NI	NC	Data area
Breastcancer	9	699	2	Biology
BreastEW	30	569	2	Biology
CongressEW	16	435	2	Politics
Exactly	13	1000	2	Biology
Exactly2	13	1000	2	Biology
HeartEW	13	270	2	Biology
IonosphereEW	34	351	2	Electromagnetic
KrvskpEW	36	3196	2	Game
Lymphography	18	148	2	Biology
M-of-n	13	1000	2	Biology
PenglungEW	325	73	2	Biology
SonarEW	60	208	2	Biology
SpectEW	22	267	2	Biology
Tic-tac-toe	9	958	2	Game
Vote	16	300	2	Politics
WaveformEW	40	5000	3	Physics
WineEW	13	178	3	Chemistry
Zoo	16	101	6	Artificial

Evaluation Measures

Each algorithm was run 30 times, the averages of the obtained results were calculated and reported in this section and compared with regard to the following evaluation measures.

- **Average Classification Accuracy**: this measure is used to calculate the accuracy of classifier produced from selected features as given in Eq. 27.

$$\text{AvgAccuracy} = \frac{1}{M}\sum_{j=1}^{M}\frac{1}{N}\sum_{i=1}^{N}(C_i == L_i) \tag{27}$$

  where M is the number of runs for the optimizer, N indicate the number of tuples in the test data, C_i and L_i represent the classifier output label and the reference of the label class of tuple i, respectively.
- **Average Fitness**: this measure is used to calculate the average of fitness overall runs, and the optimizer with the lowest value is considered the best one and is given as Eq. (28).

$$\text{AvgFitness} = \frac{1}{M}\sum_{j=1}^{M}Fit_*^i \tag{28}$$

  where M is the number of runs for the optimizer, and the Fit_*^i represents the fitness value at run number i.
- **Average selection size**: this represents the average size of the selected features to the total features number and is given as Eq. 29.

$$\text{AvgSize} = \frac{1}{M}\sum_{i=1}^{M}\frac{d_i^*}{D} \tag{29}$$

  where M is the number of runs for the optimizer, d_i^* represents the selected features number, and D represents the total number of features in the original dataset.

Table 3 The impact of the population size (N) on the accuracy results of proposed approaches

Pop size	10	20	30	40	50
	HSGW				
Dataset	Exactly				
Accuracy	**1**	**1**	**1**	**1**	**1**
Dataset	penglungEW				
Accuracy	**1**	**1**	**1**	0.94103	0.85556
Dataset	WaveformEW				
Accuracy	0.75933	**0.74647**	0.76153	0.75793	0.75547
#Wins	2	**3**	2	1	1
	RSGW				
Dataset	Exactly				
Accuracy	0.9995	**1**	**1**	**1**	**1**
Dataset	penglungEW				
Accuracy	0.93333	**1**	0.92308	**1**	0.93556
Dataset	WaveformEW				
Accuracy	0.75587	0.75857	0.75757	0.74833	0.754
#Wins	0	**3**	1	2	1
	ASGW				
Dataset	Exactly				
Accuracy	0.99983	0.99983	**1**	**1**	**1**
Dataset	penglungEW				
Accuracy	0.93333	**1**	**1**	0.86444	0.86667
Dataset	WaveformEW				
Accuracy	0.74927	0.7471	0.75283	0.765	**0.7706**
#Wins	0	1	2	1	2
#Overall wins	2	7	5	4	4

Bold marks the best results

– **Average running time**: this represents the computation time for a given optimizer in a millisecond and is calculated over the all runs and given as Eq. 30.

$$\text{AvgTime} = \frac{1}{M}\sum_{i=1}^{M} RT_i \tag{30}$$

where M is the number of runs for the optimizer, RT_i indicates the runtime in millisecond at run number i.

Parameter Tuning

The number of iterations and the population size parameters have a high impact on the performance of any optimizer; thus, we are interested in conducting an initial empirical study to assess their influence on the performance of the three proposed approaches. In Table 3, a set of different populations size has (i.e., 10, 20, 30, 40, 50) been tested, where these values are some possible settings. Then, in the next step, we conducted more experiments to assess the most suitable number of iterations (presented in Table 4), where the population size that revealed the best results has been used. Please note that three datasets with different sizes (small, medium, and large) were selected from the eighteen core datasets for testing purposes, and we reported the classification accuracy.

From Table 3, one can observe that using 20 search agents exposed the best results on the three datasets when testing the HSGW and RSGW methods, and one time when the penglungEW was used with the ASGW approach. In total, the populations of 20 obtained the best results in 7 out of 9 cases.

Table 4 presents the classification accuracy with different values of the number of iterations and 20 as a populations size. It is evident that running the algorithms for 100 iterations revealed the best performance on 6 cases out of 9. Therefore, in all subsequent experiments, the values of the population size and number of iterations parameters were used as 20 and 100, respectively.

Results and Discussion

In this subsection, the results obtained for the proposed approaches are presented and discussed in comparison with several well-established algorithms. Firstly, we compare the results of the proposed approach (HSGW) with the basic

Table 4 The impact of the number of iterations (it) on the accuracy results of proposed approaches

# iterations	50	75	100	150	200
	HSGW				
Dataset	Exactly				
Accuracy	**1**	**1**	**1**	**1**	**1**
Dataset	penglungEW				
Accuracy	**1**	0.93333	**1**	0.93333	0.88889
Dataset	WaveformEW				
Accuracy	0.75453	0.74623	0.75083	0.7638	0.76057
#Wins	2	**1**	2	1	1
	RSGW				
Dataset	Exactly				
Accuracy	0.99267	0.99917	1	1	1
Dataset	penglungEW				
Accuracy	1	1	1	0.93333	0.88889
Dataset	WaveformEW				
Accuracy	0.7618	0.75033	0.7635	0.76033	0.7485
#Wins	1	1	2	1	1
	ASGW				
Dataset	Exactly				
Accuracy	0.99683	1	1	1	1
Dataset	penglungEW				
Accuracy	1	0.93333	1	0.93333	1
Dataset	WaveformEW				
Accuracy	0.74937	0.75907	0.75697	0.76773	0.7605
#Wins	1	1	2	1	2
#Overall Wins	4	3	6	3	4

Bold marks the best results

GWO and WOA, to investigate the influence of using this hybrid model on the evaluation measures (i.e., classification accuracy, selected features number, fitness value, and the running time). After analyzing the obtained results, the performance of HSGW is compared with the state-of-the-art methods to substantiate the efficacy of this approach in tackling FS problems. It is also compared with some popular wrapper FS methods from the literature.

Results of Hybrid HSGW, RSGW, and ASGW Approaches

Here, we are interested in presenting the results of the proposed hybrid approaches and highlighting how the proposed improvements can enhance the performance of the GWO and WOA algorithms. Table 5 compares the performance of the proposed approaches in terms of classification accuracy.

Table 5 The average and standard deviation of classification accuracies of the enhanced approaches versus original methods

Dataset	Measure	bGWO	WOA	HSGW	RSGW	ASGW
Breastcancer	AVG	0.979	0.970	**0.986**	0.971	0.985
	STD	0.005	0.007	0.000	0.001	0.002
BreastEW	AVG	0.979	0.982	0.981	0.982	**1.000**
	STD	0.011	0.005	0.005	0.002	0.000
CongressEW	AVG	0.990	0.990	0.975	0.961	**0.994**
	STD	0.008	0.007	0.005	0.007	0.006
Exactly	AVG	0.900	0.832	**1.000**	0.997	0.999
	STD	0.136	0.137	0.001	0.006	0.004
Exactly2	AVG	0.752	0.709	**0.815**	0.779	0.777
	STD	0.016	0.012	0.000	0.014	0.008
HeartEW	AVG	0.847	0.802	**0.923**	0.848	0.831
	STD	0.031	0.025	0.007	0.021	0.018
IonosphereEW	AVG	0.920	0.938	0.944	**0.978**	0.972
	STD	0.023	0.018	0.008	0.007	0.009
KrvskpEW	AVG	0.958	0.942	**0.973**	0.972	0.971
	STD	0.022	0.040	0.003	0.004	0.003
Lymphography	AVG	0.921	0.851	**0.934**	0.893	0.884
	STD	0.030	0.057	0.011	0.024	0.017
M-of-n	AVG	0.950	0.945	**1.000**	1.000	**1.000**
	STD	0.065	0.067	0.000	0.001	0.000
penglungEW	AVG	0.953	0.837	0.942	**1.000**	**1.000**
	STD	0.040	0.052	0.023	0.000	0.000
SonarEW	AVG	0.975	0.919	0.964	**0.979**	0.948
	STD	0.020	0.024	0.012	0.010	0.012
SpectEW	AVG	0.812	0.812	0.862	0.815	**0.870**
	STD	0.013	0.040	0.014	0.006	0.012
Tic-tac-toe	AVG	0.831	0.789	0.828	0.859	**0.865**
	STD	0.036	0.033	0.000	0.000	0.000
Vote	AVG	0.948	0.989	0.983	**0.996**	0.984
	STD	0.011	0.013	0.003	0.007	0.003
WaveformEW	AVG	**0.766**	0.732	0.748	0.757	0.746
	STD	0.011	0.016	0.005	0.005	0.005
WineEW	AVG	**1.000**	0.978	**1.000**	**1.000**	**1.000**
	STD	0.000	0.015	0.000	0.000	0.000
Zoo	AVG	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	STD	0.000	0.000	0.000	0.000	0.000
Overall ranking	F-test	3.361	4.361	2.472	**2.361**	2.444

Bold marks the best results

Inspecting the results in Table 5, it can be observed that the hybrid approaches performed better than both GWO and WOA in all datasets except WaveformEW. The best performing approach is HSGW, which hits the highest accuracy rates in 50% of datasets. ASGW is ranked second and shows the best results in eight datasets, followed by RSGW with six datasets. Compared with enhanced variants the original algorithms can compete only in few datasets, and achieved the best results solely in dealing with the three datasets (WaveformEW, WineEW, and Zoo) and one dataset (Zoo), respectively. *F* test statistic also supports the superiority of RSGW over other competitors.

These results can be reasoned due to the capabilities of the hybrid approaches in overcoming the core searching limitation of each algorithm. In other words, hybridization helped these algorithms to escape from LO and boost the capabilities of both GWO and WOA in balancing between exploration and exploitation. Enhanced variants can inherit the main exploration merits of WOA in addition to the exploitative leanings of GWO, simultaneously.

In order to study the significance of the obtained results, the non-parametric Wilcoxon ranksum statistical test is conducted at 5% significance level for each of the proposed hybrid approaches against the basic optimizers. Table 6 shows the obtained *p* values of the test.

The ranksum test results show that the three hybrid approaches significantly outperform GWO and WOA on the majority of the datasets, which a slight advantage for HGWO and ASGW over RSGA can be seen. This indicates that the serialization approach and adaptive switching can be more efficient in terms of classification accuracy over the basic and randomized switching approaches.

The average fitness values with *F* test statistic are exposed in Table 7. As per results in Table 7, we see these results support the findings mentioned above as well. It can be observed that minimum fitness values are dominated by the hybrid approaches for all datasets except for the last three ones, which are WaveformEW, WineEW, and Zoo. *F* test statistic supports the superiority of HSGW over other competitors.

The average convergence curves for the hybrid and basic algorithms are shown for all datasets in Fig. 8. It can be observed from Fig. 8 that the hybrid approaches can show a faster convergence rate than the original algorithms in many datasets. Specifically, the curves of HSGW and ASGW managed to hit the minimum value in 77.77% of datasets. Some stagnation behaviors can also be seen in the curves of GWO and WOA in dealing with some datasets such as Breastcancer, Exactly2, HeartEW, and SpectEW.

The average number of selected features obtained by different variants versus original methods are reported in Table 8. Given selected features, it can be seen that the results are very competitive on the majority of the datasets with a slight advantage for the basic algorithms GWO and

Table 6 p-values of the Wilcoxon rank sum test (p $\leq$ 0.05 are shown in bold face, NaN: Not Applicable)

Dataset	HGW vs.		RSGW vs.		ASGW vs.	
	GWO	WOA	GWO	WOA	GWO	WOA
Breastcancer	**9.20E-09**	**8.66E-13**	**7.73E-10**	5.47E-01	**2.88E-06**	**1.79E-11**
BreastEW	7.59E-01	2.78E-01	4.25E-01	1.00E+00	**9.42E-13**	**3.76E-13**
Exactly	**1.04E-03**	**3.79E-09**	1.13E-01	**2.45E-07**	**6.67E-03**	**1.77E-08**
Exactly2	**1.14E-12**	**3.89E-13**	**3.21E-08**	**5.93E-12**	**1.06E-08**	**9.02E-12**
HeartEW	**4.53E-12**	**4.12E-12**	9.08E-01	**9.80E-09**	**1.92E-02**	**7.12E-06**
Lymphography	**1.96E-02**	**4.68E-09**	**2.60E-04**	**3.98E-04**	**1.84E-06**	**3.36E-03**
M-of-n	**1.45E-04**	**1.69E-08**	**4.68E-04**	**4.16E-08**	**1.45E-04**	**1.69E-08**
penglungEW	4.81E-01	**6.55E-11**	**2.67E-07**	**3.67E-12**	**2.67E-07**	**3.67E-12**
SonarEW	**2.20E-02**	**1.37E-09**	4.28E-01	**1.61E-11**	**3.12E-07**	**1.18E-06**
SpectEW	**9.81E-12**	**1.61E-06**	2.72E-01	7.60E-01	**7.51E-12**	**3.75E-08**
CongressEW	**6.10E-10**	**1.03E-10**	**8.39E-12**	**1.06E-11**	9.96E-02	**3.61E-02**
IonosphereEW	**3.56E-07**	1.17E-01	**2.54E-11**	**1.19E-10**	**5.85E-11**	**8.62E-10**
KrvskpEW	**2.22E-06**	**6.99E-06**	**8.63E-05**	**3.35E-05**	**7.26E-04**	**1.37E-04**
Tic-tac-toe	6.02E-01	**1.89E-07**	5.74E-02	**1.19E-12**	5.74E-02	**1.19E-12**
Vote	**8.11E-13**	**1.54E-03**	**3.77E-12**	**2.39E-02**	**6.47E-13**	**5.32E-03**
WaveformEW	**2.69E-09**	**1.40E-05**	**1.70E-03**	**3.54E-09**	**4.90E-10**	**1.03E-04**
WineEW	NaN	**7.25E-09**	NaN	**7.25E-09**	NaN	**7.25E-09**
Zoo	NaN	NaN	NaN	NaN	NaN	NaN

Table 7 The average and standard deviation values of the fitness values for the enhanced approaches versus the original methods

Dataset	Measure	bGWO	WOA	HSGW	RSGW	ASGW
Breastcancer	AVG	0.026	0.035	**0.020**	0.036	0.021
	STD	0.005	0.007	0.000	0.001	0.002
BreastEW	AVG	0.024	0.023	0.025	0.023	**0.005**
	STD	0.011	0.005	0.005	0.002	0.001
CongressEW	AVG	0.013	0.012	0.031	0.046	**0.012**
	STD	0.007	0.008	0.004	0.006	0.005
Exactly	AVG	0.104	0.172	**0.006**	0.009	0.007
	STD	0.135	0.134	0.001	0.006	0.004
Exactly2	AVG	0.250	0.292	**0.191**	0.226	0.227
	STD	0.015	0.012	0.000	0.014	0.007
HeartEW	AVG	0.156	0.202	**0.084**	0.156	0.172
	STD	0.031	0.025	0.007	0.021	0.018
IonosphereEW	AVG	0.081	0.065	0.061	**0.028**	0.033
	STD	0.022	0.018	0.008	0.007	0.008
KrvskpEW	AVG	0.045	0.063	**0.034**	0.035	0.036
	STD	0.022	0.039	0.003	0.004	0.003
Lymphography	AVG	0.082	0.152	**0.071**	0.112	0.121
	STD	0.029	0.056	0.011	0.023	0.017
M-of-n	AVG	0.055	0.061	**0.006**	0.006	0.006
	STD	0.065	0.066	0.000	0.001	0.001
penglungEW	AVG	0.047	0.163	0.062	0.006	**0.005**
	STD	0.040	0.052	0.022	0.000	0.000
SonarEW	AVG	0.027	0.084	0.041	**0.027**	0.058
	STD	0.019	0.023	0.012	0.010	0.011
SpectEW	AVG	0.188	0.189	0.141	0.189	**0.134**
	STD	0.013	0.040	0.014	0.006	0.012
Tic-tac-toe	AVG	0.175	0.216	0.179	0.148	**0.143**
	STD	0.035	0.032	0.000	0.000	0.000
Vote	AVG	0.055	0.014	0.022	**0.010**	0.022
	STD	0.011	0.014	0.003	0.007	0.003
WaveformEW	AVG	**0.235**	0.271	0.257	0.248	0.258
	STD	0.011	0.015	0.005	0.005	0.005
WineEW	AVG	**0.004**	0.027	0.004	0.005	0.005
	STD	0.000	0.015	0.001	0.001	0.001
Zoo	AVG	**0.001**	0.003	0.004	0.004	0.005
	STD	0.001	0.001	0.001	0.001	0.000
Overall ranking	F-test	3.028	4.194	**2.472**	2.667	2.639

Bold marks the best results

WOA. *F* test statistic supports the relative advantage of bGWO over other competitors in terms of selected features.

Another critical aspect of the comparison is the running time. Table 9 reports the average running time in seconds for the primary and hybrid approaches on all datasets, which are computed over 30 independent runs. As a natural side effect, hybridization of metaheuristic methods with their variants will increase the computational time of the overall algorithm. For example, in the serial approach, like in HSGW, the computational time increases because there are more embedded operators in the main loop of the algorithm compared with GWO and WOA. On the other hand, in RSGW and ASGWO, there is a communication overhead resulting from switching between the execution of the algorithmic operators and exchanging the information. However, the problem of the computation time in the latter

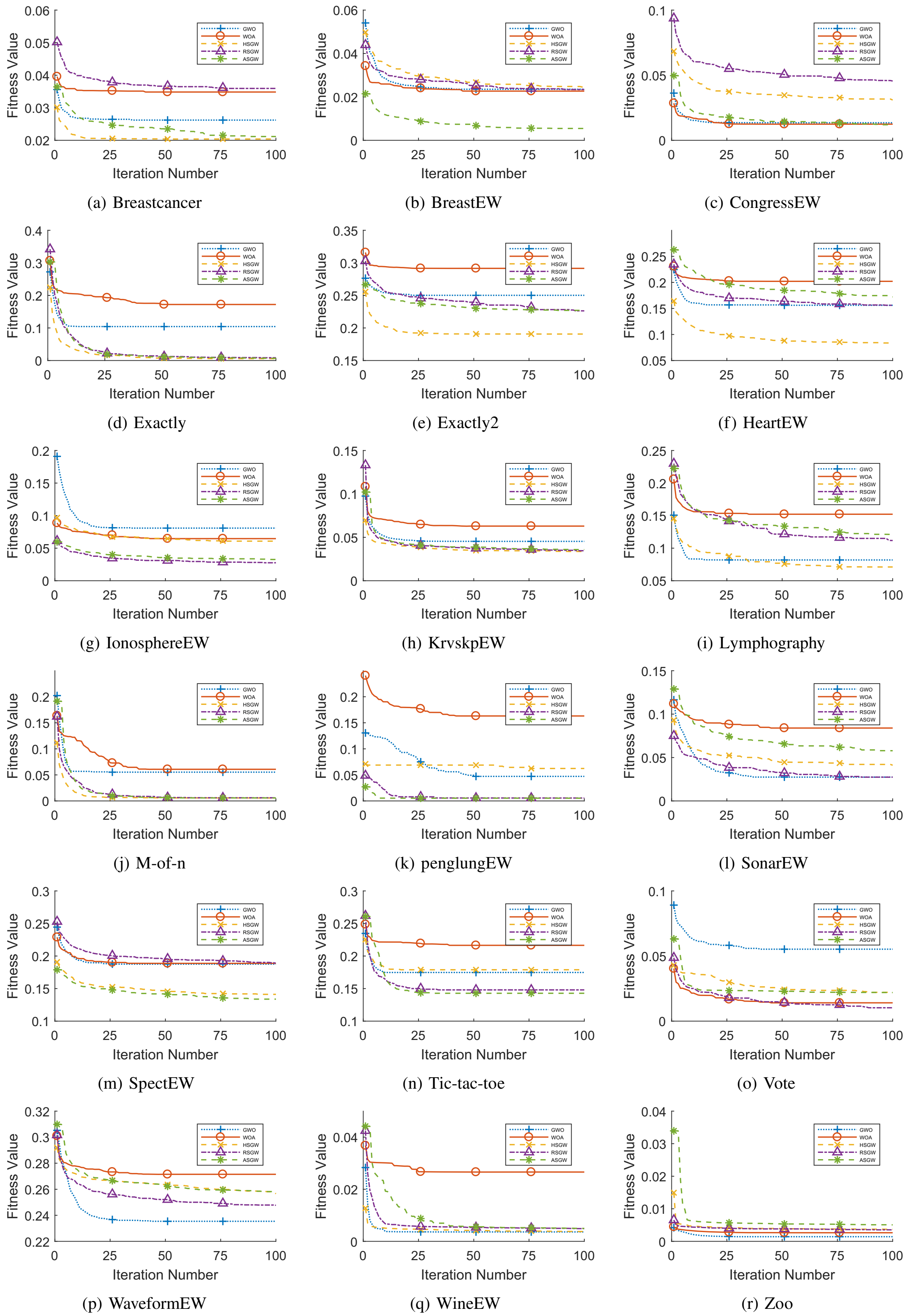

Fig. 8 Convergence curves for all compared approaches

Table 8 The average and standard deviation values of number of selected features for all approaches

Dataset	Measure	bGWO	WOA	HSGW	RSGW	ASGW
Breastcancer	AVG	4.567	**4.300**	5.000	5.933	4.867
	STD	1.006	0.750	0.000	0.365	0.346
BreastEW	AVG	**7.800**	15.400	16.667	17.500	15.833
	STD	2.091	4.256	1.561	2.080	1.821
CongressEW	AVG	6.000	**3.833**	8.867	9.700	8.833
	STD	1.948	3.086	1.224	1.622	1.895
Exactly	AVG	**5.833**	6.800	6.700	7.100	6.867
	STD	1.206	3.428	0.466	0.759	0.571
Exactly2	AVG	5.167	**4.167**	9.033	9.200	7.933
	STD	2.151	3.914	0.183	0.664	2.664
HeartEW	AVG	**5.433**	7.067	8.767	6.133	6.367
	STD	1.006	1.701	0.568	1.279	1.426
IonosphereEW	AVG	**5.967**	10.933	18.167	20.500	17.300
	STD	1.426	4.913	2.230	2.432	2.231
KrvskpEW	AVG	**12.333**	20.500	24.800	24.800	24.500
	STD	2.123	4.447	1.937	2.140	1.796
Lymphography	AVG	**6.433**	8.067	10.567	10.567	11.200
	STD	1.382	2.363	2.402	1.654	1.730
M-of-n	AVG	**6.500**	7.967	6.800	7.100	6.867
	STD	0.820	1.299	0.484	0.548	0.629
penglungEW	AVG	**15.200**	46.233	165.533	181.200	170.300
	STD	3.624	17.741	17.324	15.057	14.295
SonarEW	AVG	**13.533**	22.600	34.300	36.433	35.300
	STD	2.649	6.547	4.308	4.049	3.456
SpectEW	AVG	**3.767**	6.233	10.233	13.300	10.167
	STD	3.339	3.202	2.079	1.878	1.555
Tic-tac-toe	AVG	5.867	**5.800**	7.000	7.000	7.000
	STD	0.973	0.887	0.000	0.000	0.000
Vote	AVG	5.333	**4.600**	7.567	8.800	8.967
	STD	2.845	3.369	1.251	1.518	0.765
WaveformEW	AVG	**14.833**	23.767	26.933	27.533	25.833
	STD	2.692	6.719	2.677	2.360	2.350
WineEW	AVG	**4.333**	5.533	4.533	5.867	5.933
	STD	0.547	1.502	0.629	0.730	1.337
Zoo	AVG	**2.133**	3.967	5.533	5.300	7.600
	STD	1.106	1.650	0.776	0.915	0.675
Overall ranking	F-test	**1.278**	2.111	3.556	4.389	3.667

Bold marks the best results

approach is less, because only one of the algorithms runs at a time in each iteration, whereas both algorithms run in each iteration in the serial version. Inspecting the results in Table 9, we can see that the recorded values are in line with the discussed logic. GWO and WOA are the fastest with a very competitive average running time. *F* test statistic supports the superiority of binary WOA over bGWO. As predicted, RSGW and ASGW come next with a slight increase in the average running time of less than 1 second on the majority of the cases. This leads to a more accurate set of features and eventually, a more accurate classification.

It is worth noting that HSGW is the slowest approach, which takes almost the double running time compared with GWO and WOA. Taking both accuracy rates and

Table 9 The average computational time and the StDev values for the enhanced approaches versus the original methods

Dataset	Measure	bGWO	WOA	HSGW	RSGW	ASGW
Breastcancer	AVG	10.350	**9.492**	21.240	10.353	10.185
	STD	0.363	0.541	0.874	0.445	0.573
BreastEW	AVG	9.956	**9.099**	19.485	9.729	9.515
	STD	0.606	0.542	1.114	0.373	0.656
CongressEW	AVG	8.887	**8.311**	18.321	8.877	8.704
	STD	0.628	0.413	1.022	0.599	0.399
Exactly	AVG	**11.279**	11.911	25.009	12.870	12.087
	STD	0.529	0.959	1.111	0.589	0.595
Exactly2	AVG	11.084	**10.772**	26.035	13.080	13.075
	STD	0.712	1.169	0.960	0.538	0.835
HeartEW	AVG	8.228	**8.040**	16.884	8.492	8.301
	STD	0.577	0.416	0.636	0.366	0.350
IonosphereEW	AVG	9.111	**8.093**	17.024	8.557	8.383
	STD	0.552	0.602	1.022	0.441	0.405
KrvskpEW	AVG	**55.505**	67.574	164.317	81.422	79.977
	STD	5.708	7.046	5.002	2.329	3.594
Lymphography	AVG	8.013	7.620	16.174	7.920	**7.542**
	STD	0.488	0.542	1.023	0.338	0.316
M-of-n	AVG	**10.980**	11.527	24.645	12.965	12.081
	STD	0.756	0.695	0.852	0.996	0.469
penglungEW	AVG	**7.942**	9.143	20.466	9.758	9.265
	STD	0.570	0.430	0.950	0.616	0.602
SonarEW	AVG	7.731	**7.627**	16.620	8.321	8.539
	STD	0.397	0.353	1.018	0.506	0.350
SpectEW	AVG	8.137	7.996	15.991	8.042	**7.885**
	STD	0.397	0.553	0.772	0.395	0.375
Tic-tac-toe	AVG	10.890	**10.852**	24.190	11.873	11.586
	STD	0.519	0.696	1.065	0.613	0.604
Vote	AVG	8.392	**7.908**	16.829	8.417	8.188
	STD	0.372	0.536	0.653	0.270	0.536
WaveformEW	AVG	**133.433**	170.281	409.805	203.854	197.923
	STD	12.730	25.266	9.388	6.961	6.485
WineEW	AVG	7.870	**7.486**	16.393	8.054	7.892
	STD	0.583	0.427	0.691	0.454	0.494
Zoo	AVG	7.682	**7.453**	15.798	7.929	7.781
	STD	0.324	0.348	0.756	0.554	0.381
Overall ranking	F-test	2.389	**1.389**	5	3.667	2.556

Bold marks the best results

the running times into consideration, the proposed ASGA proves to be much preferable because it achieved high accuracy in classification and very competitive running time simultaneously.

Figures 9 and 10 also demonstrate the boxplots of accuracy rates for proposed hybrid optimizers versus the bGWO and WOA in tackling all datasets. The superiority of the proposed hybrids are evident on the majority of data sets.

Comparison with State-of-the-Art Algorithms

In this subsection, the efficacy of the proposed approaches is compared to four state-of-the-art algorithms from literature.

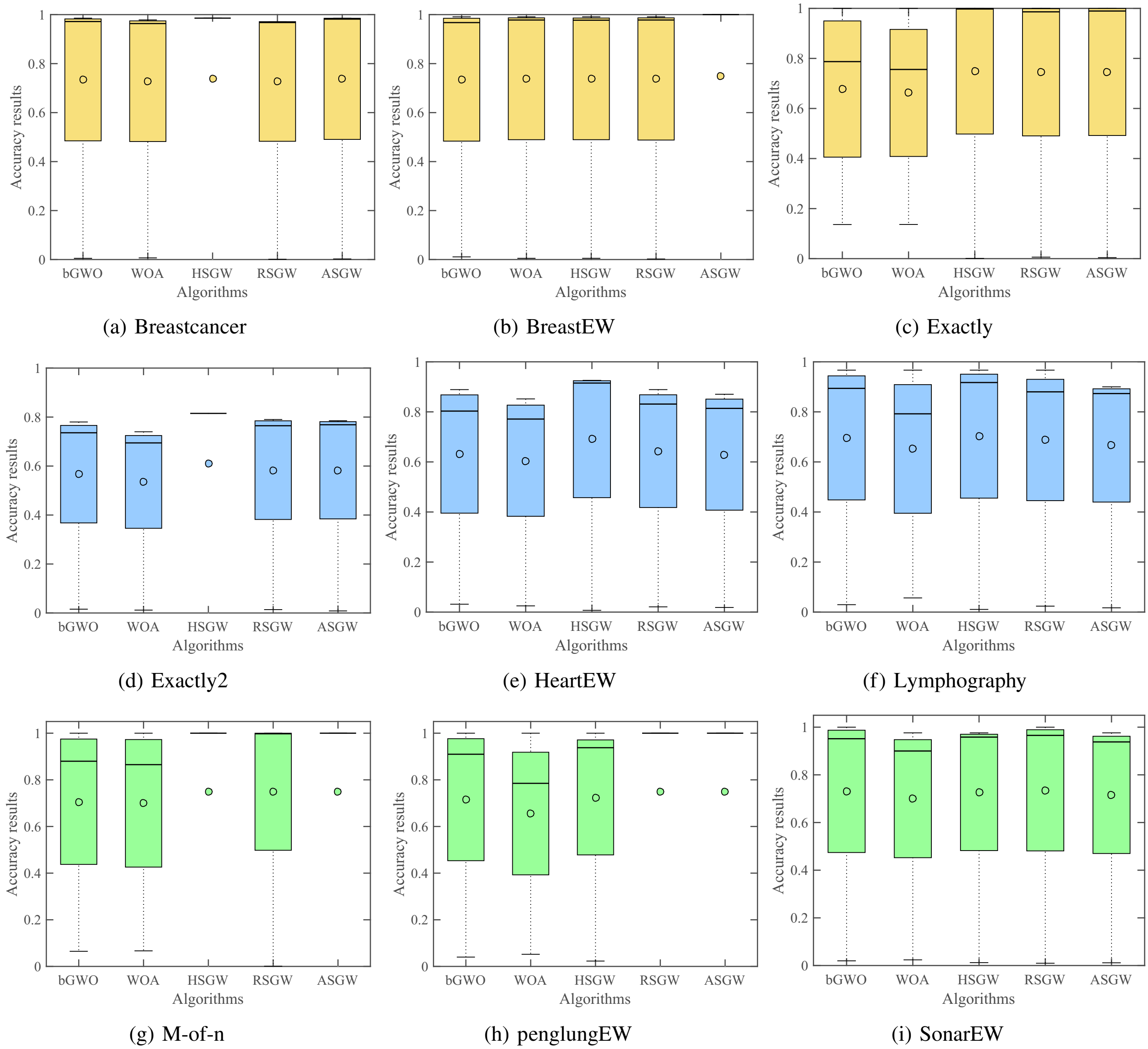

Fig. 9 Boxplots of accuracy rates for HSGW, RSGW, and ASGW compared to bGWO and WOA optimizers in dealing with Breastcancer, BreastEW, Exactly, Exactly2, HeartEW, Lymphography, M-of-n, penglungEW, and SonarEW datasets

The latter wrapper FS methods are binary grasshopper algorithm (BGOA), binary gravitational search algorithm (BGSA), GA, and binary PSO (BPSO). As previous tests, the comparison is made based on the average classification accuracy, number of selected features, and values of the fitness.

Average accuracy rates are shown in Fig. 11 and presented in Table 10. The best results are highlighted in boldface. We can observe that the proposed approaches have achieved the highest accuracy rates on the majority of datasets. HSGW provides the best results on eight (44.44%) datasets, while each of ASGW and RSGW hit the highest rates on seven (38.88%) datasets. On the other hand, GA and BPSO managed to find the fittest results on only two (11.11%) datasets. The average fitness values for all studied methods are also reported in Table 11. These values confirm the superiority of the hybrid approaches because, in majority of the dataset, the minimum value was reached by one of the approaches. F test statistic shows the superiority of HSGW over other competitors.

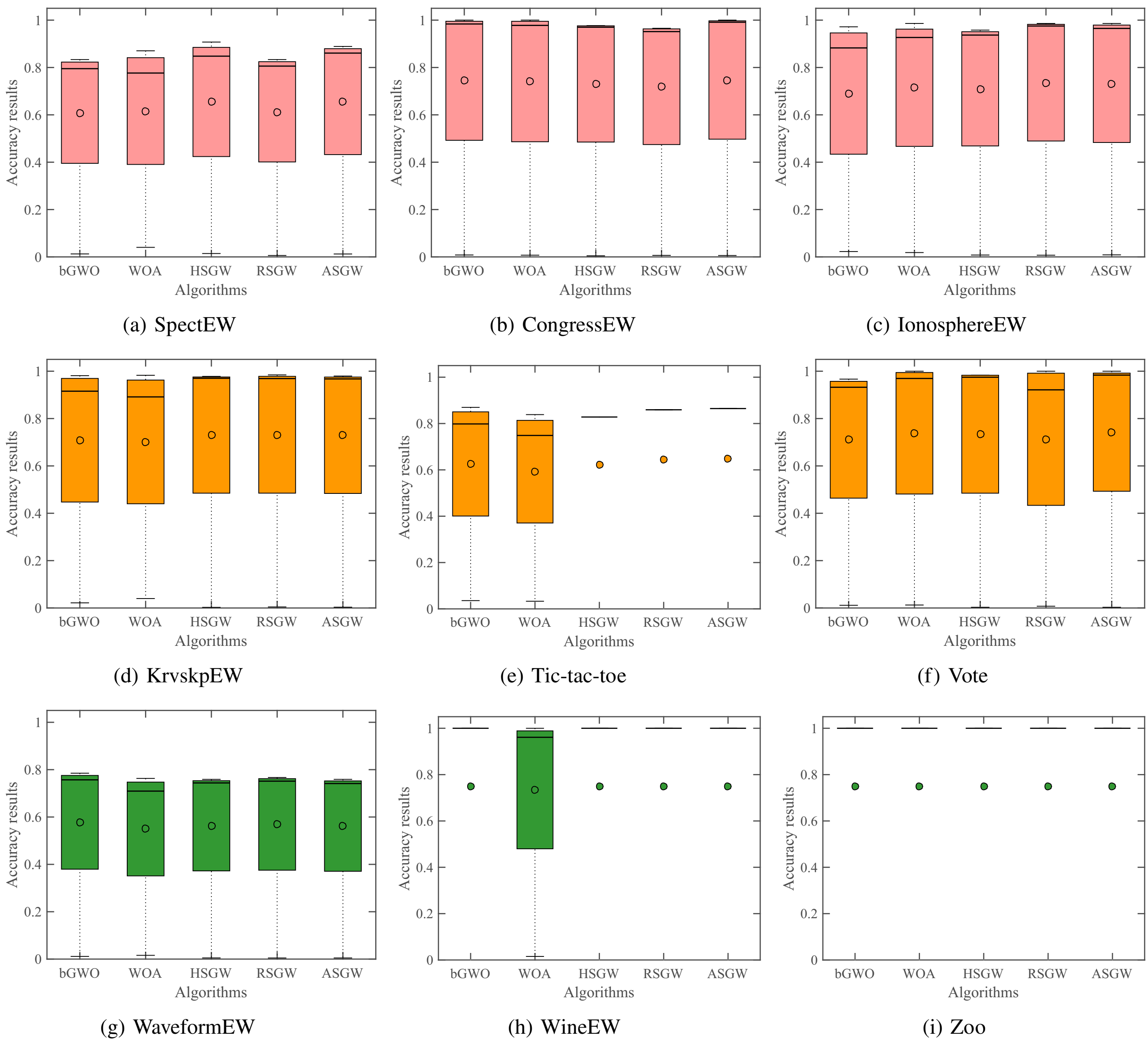

Fig. 10 Boxplots of accuracy rates for HSGW, RSGW, and ASGW compared to bGWO and WOA optimizers in dealing with SpectEW, CongressEW, IonosphereEW, KrvskpEW, Tic-tac-toe, Vote, WaveformEW, WineEW, and Zoo datasets

Considering the number of selected features, Table 12 shows the average results for all datasets. Inspecting the results in Table 12, GA has obtained the best results by selecting the smallest number of features on 11 datasets. However, the proposed approaches are very competitive in the majority of cases.

Comparison with Literature

To further verify the performance of the proposed algorithms, the results are compared with those reported in well-established literature on the same benchmark datasets. The compared methods are two variations of each of GA, PSO and bGWO reported in [9, 28]. The results of the compared methods and the proposed approaches are given in Table 13.

As per results in Table 13, the superiority of proposed approaches can be observed for 83.33 % of datasets. Inspecting the results in Table 13, it can be seen that the HSGA has the highest accuracy rate on nine (50%) datasets, followed by ASGA with better performance on seven datasets, and RSGW with leading results on five datasets. On other hand, the compared methods only hit the highest rates in tackling three datasets. Additionally, *F* test results shows that the HSGW has attained the best rank compared to other peer.

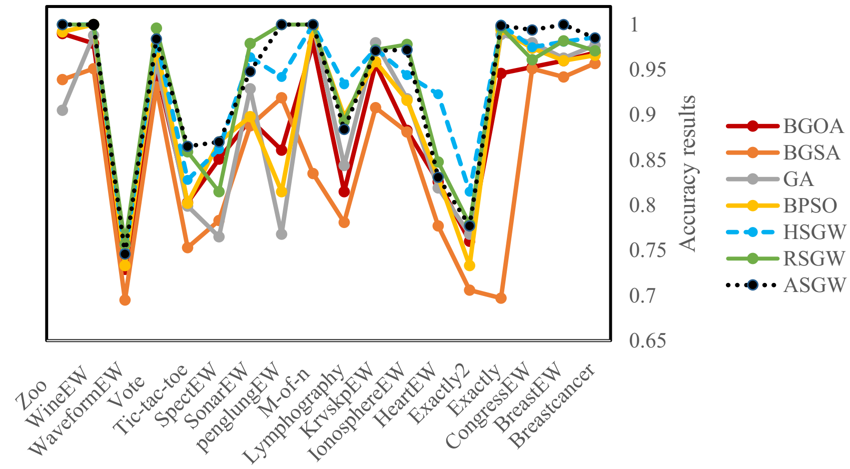

Fig. 11 Monitoring of accuracy rates attained by the proposed HSGW, RSGW, ASGW methods compared to those obtained by BGOA, BGSA, GA, and BPSO wrapper FS algorithms

Figure 12 builds a surface of accuracy rates to visually compare the accuracy results of proposed approaches versus those from literature. From Fig. 12, we can detect that the classification results of the HSGW, RSGW, and ASGW tend to be superior as compared with other methods because of darker colors. The golden regions show that the accuracy of methods is inside [60%, 80%] interval. It is seen that the proposed variants have smaller golden zones and more darker black regions in addition to some green parts (100 % accuracy), which visually indicates the improvements of the results.

To sum up, proposed hybrid approaches outperform all other approaches on the majority of the datasets considering all evaluation criteria. Out of three proposed approaches, it can be concluded that ASGW is more advantageous because it demonstrated superior results in terms of average accuracy rates, and very competitive running time compared to the primary approaches. This fact shows that simultaneous use of the operator of two algorithms in a hybrid is better than using the sequentially in term of finding more accurate solutions. The run time of simultaneous hybrids tends to be less than sequential ones as well. This is perhaps due to possible undesired damages to a population when applying an algorithm after the other. Although the results show that RSGW and HSGW are comparative as compared with other similar algorithms, the quality of a population might

Table 10 Average classification accuracy for the proposed HSGW, RSGW, and ASGW approaches versus the state-of-the-art BGOA, BGSA, GA, and BPSO wrapper FS methods

Dataset	BGOA	BGSA	GA	BPSO	HSGW	RSGW	ASGW
Breastcancer	0.969	0.957	0.977	0.966	**0.986**	0.971	0.985
BreastEW	0.960	0.942	0.963	0.960	0.981	0.982	**1.000**
CongressEW	0.953	0.951	0.980	0.974	0.975	0.961	**0.994**
Exactly	0.946	0.697	0.989	0.994	**1.000**	0.997	0.999
Exactly2	0.760	0.706	0.768	0.733	**0.815**	0.779	0.777
HeartEW	0.826	0.777	0.819	0.831	**0.923**	0.848	0.831
IonosphereEW	0.883	0.881	0.918	0.916	0.944	**0.978**	0.972
KrvskpEW	0.956	0.908	**0.980**	0.959	0.973	0.972	0.971
Lymphography	0.815	0.781	0.844	0.898	**0.934**	0.893	0.884
M-of-n	0.979	0.835	**1.000**	0.997	**1.000**	1.000	**1.000**
penglungEW	0.861	0.919	0.768	0.815	0.942	**1.000**	**1.000**
SonarEW	0.895	0.888	0.929	0.898	0.964	**0.979**	0.948
SpectEW	0.851	0.783	0.765	**0.870**	0.862	0.815	0.870
Tic-tac-toe	0.803	0.753	0.799	0.802	0.828	0.859	**0.865**
Vote	0.951	0.931	0.959	0.977	0.983	**0.996**	0.984
WaveformEW	0.729	0.695	0.756	0.733	0.748	**0.757**	0.746
WineEW	0.979	0.951	0.988	**1.000**	**1.000**	**1.000**	**1.000**
Zoo	0.990	0.939	0.905	0.992	**1.000**	**1.000**	**1.000**
F test	5.472	6.722	4.417	4.389	**2.222**	2.417	2.361

Table 11 Average fitness values for the proposed HSGW, RSGW, and ASGW approaches versus the state-of-the-art BGOA, BGSA, GA, and BPSO wrapper FS methods

Dataset	BGOA	BGSA	GA	BPSO	HGW	RSGW	ASGW
Breastcancer	0.036	0.049	0.028	0.037	**0.020**	0.036	0.021
BreastEW	0.045	0.063	0.040	0.045	0.025	0.023	**0.005**
CongressEW	0.049	0.053	0.023	0.028	0.031	0.046	**0.012**
Exactly	0.059	0.307	0.016	0.011	**0.006**	0.009	0.007
Exactly2	0.239	0.295	0.230	0.268	**0.191**	0.226	0.227
HeartEW	0.178	0.226	0.183	0.172	**0.084**	0.156	0.172
IonosphereEW	0.120	0.122	0.085	0.087	0.061	**0.028**	0.033
KrvskpEW	0.049	0.097	**0.024**	0.046	0.034	0.035	0.036
Lymphography	0.187	0.222	0.159	0.106	**0.071**	0.112	0.121
M-of-n	0.027	0.170	**0.005**	0.008	0.006	0.006	0.006
penglungEW	0.142	0.085	0.233	0.187	0.062	0.006	**0.005**
SonarEW	0.109	0.116	0.075	0.106	0.041	**0.027**	0.058
SpectEW	0.152	0.220	0.236	**0.133**	0.141	0.189	0.134
Tic-tac-toe	0.203	0.251	0.207	0.202	0.179	0.148	**0.143**
Vote	0.052	0.073	0.044	0.027	0.022	**0.010**	0.022
WaveformEW	0.274	0.307	**0.247**	0.270	0.257	0.248	0.258
WineEW	0.025	0.054	0.016	0.006	**0.004**	0.005	0.005
Zoo	0.015	0.065	0.098	0.013	0.004	**0.004**	0.005
F test	5.500	6.722	4.278	4.333	**2.167**	2.5278	2.472

Bold marks the best results

Table 12 Average number of selected features for the proposed HSGW, RSGW, and ASGW approaches versus the state-of-the-art BGOA, BGSA, GA, and BPSO wrapper FS methods

Dataset	BGOA	BGSA	GA	BPSO	HGW	RSGW	ASGW
Breastcancer	4.000	6.067	5.000	**3.100**	5.000	5.933	4.867
BreastEW	15.367	16.567	**12.000**	16.467	16.667	17.500	15.833
CongressEW	4.333	6.767	4.933	**4.033**	8.867	9.700	8.833
Exactly	7.633	8.733	**5.933**	6.600	6.700	7.100	6.867
Exactly2	1.267	5.100	**1.000**	4.767	9.033	9.200	7.933
HeartEW	6.767	6.833	**5.667**	6.733	8.767	6.133	6.367
IonosphereEW	13.433	15.400	**12.600**	13.600	18.167	20.500	17.300
KrvskpEW	19.900	19.967	**15.767**	19.300	24.800	24.800	24.500
Lymphography	**7.467**	9.167	7.900	9.500	10.567	10.567	11.200
M-of-n	7.533	8.467	**6.000**	6.867	6.800	7.100	6.867
penglungEW	150.133	157.167	**109.867**	147.900	165.533	181.200	170.300
SonarEW	28.567	30.033	**25.600**	30.833	34.300	36.433	35.300
SpectEW	9.933	9.533	**7.400**	9.100	10.233	13.300	10.167
Tic-tac-toe	6.833	5.867	7.000	**5.133**	7.000	7.000	7.000
Vote	**5.267**	8.167	5.800	5.800	7.567	8.800	8.967
WaveformEW	21.200	**19.900**	20.100	22.000	26.933	27.533	25.833
WineEW	6.333	7.367	6.400	7.800	**4.533**	5.867	5.933
Zoo	8.133	8.167	**5.200**	8.033	5.533	5.300	7.600
F test	3.167	4.556	**1.972**	3.056	4.889	5.639	4.722

Bold marks the best results

Table 13 Classification accuracies of the proposed HSGW, RSGW, and ASGW algorithms compared with other optimizers from literature

Datasets	GA[28]	PSO[28]	bGWO1[9]	bGWO2 [9]	GA1 [9]	PSO1 [9]	HSGW	RSGW	ASGW
Breastcancer	0.957	0.949	0.976	0.975	0.968	0.967	**0.986**	0.971	0.985
BreastEW	0.923	0.933	0.924	0.935	0.939	0.933	0.981	0.982	**1.000**
Exactly	0.822	0.973	0.708	0.776	0.674	0.688	0.975	0.961	**0.994**
Exactly2	0.677	0.666	0.745	0.750	0.746	0.730	**1.000**	0.997	0.999
HeartEW	0.732	0.745	0.776	0.776	0.780	0.787	**0.815**	0.779	0.777
Lymphography	0.758	0.759	0.744	0.700	0.696	0.744	**0.923**	0.848	0.831
M-of-n	0.916	**0.996**	0.908	0.963	0.861	0.921	0.944	0.978	0.972
penglungEW	0.672	0.879	0.600	0.584	0.584	0.584	**0.973**	0.972	0.971
SonarEW	0.833	0.804	0.731	0.729	0.754	0.737	**0.934**	0.893	0.884
SpectEW	0.756	0.738	0.820	0.822	0.793	0.822	**1.000**	1.000	**1.000**
CongressEW	0.898	0.937	0.935	0.938	0.932	0.928	0.942	**1.000**	**1.000**
IonosphereEW	0.863	0.876	0.807	0.834	0.814	0.819	0.964	**0.979**	0.948
KrvskpEW	0.940	0.949	0.944	**0.956**	0.920	0.941	0.862	0.815	0.870
Tic-tac-toe	0.764	0.750	0.728	0.727	0.719	0.735	0.828	0.859	**0.865**
Vote	0.808	0.888	0.912	0.920	0.904	0.904	0.983	**0.996**	0.984
WaveformEW	0.712	0.732	0.786	**0.789**	0.773	0.762	0.748	0.757	0.746
WineEW	0.947	0.937	0.930	0.920	0.937	0.933	**1.000**	**1.000**	**1.000**
Zoo	0.946	0.963	0.879	0.879	0.855	0.861	**1.000**	**1.000**	**1.000**
W|T|L	0|0|18	1|0|17	0|0|18	2|0|16	0|0|18	0|0|18	9|0|9	5|0|13	6|0|13
F test	6.556	5.556	6.194	5.472	6.667	6.278	**2.611**	2.861	2.806

Bold marks the best results

be endangered when applying a different algorithm with a different search mechanism iteratively without considering the changes done by the first algorithm. That might also interrupt the search pattern of either of the algorithms too. On the other hand, two algorithms exchange information about the search space and work together in parallel when applying them to a population simultaneously. This was

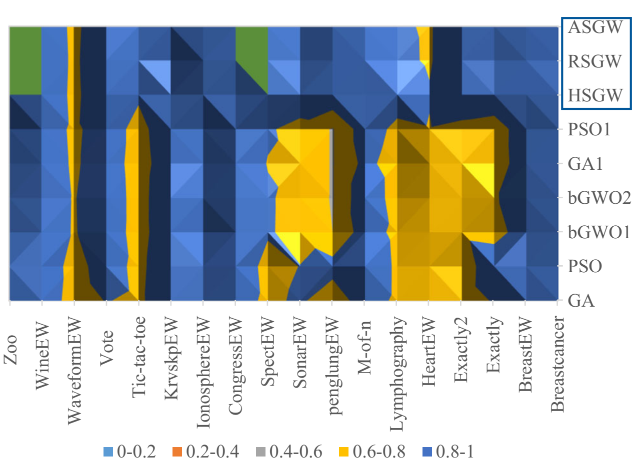

Fig. 12 Comparison of the hybrid variants versus results from literature according to the accuracy surfaces. In this chart, better results are shown with darker black and green

the case in HSGWO, which is one of the reasons why it outperformed RSGW and HSGW.

The main reason for the superior efficacy of the proposed approaches is that the improved variants can alleviate the stagnation problems and inertia to LO for each basic algorithm (i.e., GWO and WOA). In each generation of the optimization process, each individual in the population is updated using the GWO's operators; then, the same population is updated by using the WOA. The process is kept continuous since we used to employ the alpha position in the GWO stage as the leader of the WOA stage. In this way, we move the population into another potential region in the search space to find more promising regions. The benefits of this mechanism are twofold. For one, it assists the algorithm in discovering better solutions than the current ones. For another, it helps the algorithm to get away from the LO and mitigate stagnation problems. The results obtained in this paper can reveal the enhanced capabilities of the proposed approaches in demonstrating better performance than applying each individual optimizer. The observations show the abilities of the proposed variants to deeply explore the feature space for finding more promising areas, and then exploiting the explored solutions. This fact requires the proposed algorithm to discover high-quality solutions with the most informative features in dealing with different datasets.

Conclusion and Future Directions

In this research, we proposed variants of efficient hybrid meta-heuristic approaches based on two recent metaheuristics (GWO and WOA) to tackle FS problems. Even though GWO and WOA are two recently proposed optimizers, researchers applied these biology-based techniques to various problems, and both methods have shown competitive results in solving a wide range of optimization problems. Similarly to other optimizers, each algorithm has its advantages and disadvantages, i.e., GWO updates the position vector of all wolves based on the position of three fittest agents in the population. This mechanism may mislead the solutions to move towards locally optimal solutions. By contrast, a random factor is employed in WOA when moving some individuals in the feature space. This factor helps the WOA algorithm to escape from LO randomly, but it may cause a premature convergence. These problems motivated our attempts to propose three hybrid models (called HSGW, RSGW, and ASGWO) to aggregate the potential exploratory and exploitative merits of both methods. The proposed method was tested on several UCI datasets, and the results were compared in three phases. In the first phase, proposed HSGW, RSGW, and ASGWO methods were compared to the individual optimizers (i.e., GWO and WOA), then, in the second phase. A comparison with the state-of-the-art FS approaches GA, BPSO, BGSA, and BGOA was presented, and finally, the results of HSGW, RSGW, and ASGWO were compared with those from well-established literature. The results showed the superiority of HSGW over all other approaches, which proved the hypothesis of this research.

For future work, it is recommended to maintain the multi-objective formulation of the FS problems and develop a posteriori multi-objective optimization algorithm. This direction will allow finding the Pareto optimal solution set for FS problems representing the best trade-offs between the two conflicting objectives.

Funding This research was supported by the research committee at Birzeit University with a grant number 250177.

Compliance with Ethical Standards

Conflict of interests The authors declare that they have no conflict of interest.

Informed Consent All procedures followed were in accordance with the ethical standards of the responsible committee on human experimentation (institutional and national) and with the Helsinki Declaration of 1975, as revised in 2008 (5).

Human and Animal Rights This article does not contain any studies with human or animal subjects performed by any of the authors.

References

1. Aljarah I, Ala'M AZ, Faris H, Hassonah MA, Mirjalili S, Saadeh H. Simultaneous feature selection and support vector machine optimization using the grasshopper optimization algorithm. Cogn Comput. 2018;10:1–18.
2. Aljarah I, Mafarja M, Heidari AA, Faris H, Zhang Y, Mirjalili S. Asynchronous accelerating multi-leader salp chains for feature selection. Appl Soft Comput. 2018;71:964–79.
3. Aljarah I, Mafarja M, Heidari AA, Faris H, Mirjalili S. Clustering analysis using a novel locality-informed grey wolf-inspired clustering approach. Knowl Inf Syst. 2019. https://doi.org/10.1007/s10115-019-01358-x.
4. Chen H, Jiao S, Heidari AA, Wang M, Chen X, Zhao X. An opposition-based sine cosine approach with local search for parameter estimation of photovoltaic models. Energy Convers Manag. 2019;195:927–942.
5. Dash M, Liu H. Feature selection for classification. Intelligent Data Analysis. 1997;1(3):131–56.
6. Dorigo M, Di Caro G. Ant colony optimization: a new meta-heuristic. In: Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406). IEEE; 1999. p. 1470–77.
7. Eberhart R, Kennedy J. A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science, 1995. MHS'95. IEEE; 1995. p. 39–43.
8. Emary E, Zawbaa HM, Hassanien AE. Binary ant lion approaches for feature selection. Neurocomputing. 2016;213:54–65.
9. Emary E, Zawbaa HM, Hassanien AE. Binary grey wolf optimization approaches for feature selection. Neurocomputing. 2016;172:371–81.
10. Emary E, Zawbaa HM, Hassanien AE. Binary grey wolf optimization approaches for feature selection. Neurocomputing. 2016;172:371–81.
11. Faris H, Aljarah I, et al. Optimizing feedforward neural networks using krill herd algorithm for e-mail spam detection. In: 2015 IEEE Jordan conference on applied electrical engineering and computing technologies (AEECT). IEEE; 2015. P. 1–5.
12. Faris H, Aljarah I, Al-Madi N, Mirjalili S. Optimizing the learning process of feedforward neural networks using lightning search algorithm. Int J Artif Intell Tools. 2016;25(06):1650033.
13. Faris H, Aljarah I, Mirjalili S. Evolving radial basis function networks using moth–flame optimizer. Handbook of Neural Computation. Elsevier; 2017. P. 537–550.
14. Faris H, Al-Zoubi AM, Heidari AA, Aljarah I, Mafarja M, Hassonah MA, Fujita H. An intelligent system for spam detection and identification of the most relevant features based on evolutionary random weight networks. Information Fusion. 2019;48:67–83. https://doi.org/10.1016/j.inffus.2018.08.002.
15. Faris H, Mafarja M, Heidari AA, Aljarah I, Ala'M A. Z., Mirjalili S., Fujita H. An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. Knowl-Based Syst. 2018;154:43–67.
16. Fayyad U, Piatetsky-Shapiro G, Smyth P. From data mining to knowledge discovery in databases. AI Mag. 1996;17(3):37.
17. Gao W, Wu H, Siddiqui MK, Baig AQ. Study of biological networks using graph theory. Saudi J Biol Sci. 2018;25(6):1212–19.
18. Gao W, Guirao JLG, Basavanagoud B, Wu J. Partial multi-dividing ontology learning algorithm. Inf Sci. 2018;467:35–58.
19. Gao W, Wang W, Dimitrov D, Wang Y. Nano properties analysis via fourth multiplicative ABC indicator calculating. Arab J Chem. 2018;11(6):793–801.

20. Gao W, Guirao JLG, Abdel-Aty M, Xi W. An independent set degree condition for fractional critical deleted graphs. Discrete & Continuous Dynamical Systems-Series S. 2019;12(4&5):877–86.
21. Gao W, Dimitrov D, Abdo H. Tight independent set neighborhood union condition for fractional critical deleted graphs and ID deleted graphs. Discrete & Continuous Dynamical Systems-Series S. 2019;12(4&5):711–21.
22. Ghatasheh N, Faris H, Aljarah I, Al-Sayyed RMH. Optimizing software effort estimation models using firefly algorithm. arXiv:1903.02079. 2019.
23. Guyon I, Elisseeff A. An introduction to variable and feature selection. J Mach Learn Res. 2003;3(Mar):1157–82.
24. Heidari AA, Aljarah I, Faris H, Chen H, Luo J, Mirjalili S. An enhanced associative learning-based exploratory whale optimizer for global optimization. Neural Comput & Applic. 2019;1–27.
25. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H. Harris hawks optimization: algorithm and applications. Futur Gener Comput Syst. 2019;97:849–72.
26. Holland JH. Genetic algorithms. Scientific American. 1992;267(1): 66–73.
27. Jadhav AN, Gomathi N. Wgc: hybridization of exponential grey wolf optimizer with whale optimization for data clustering. Alex Eng J. 2017.
28. Kashef S, Nezamabadi-pour H. An advanced aco algorithm for feature subset selection. Neurocomputing. 2015;147:271–9. https://doi.org/10.1016/j.neucom.2014.06.067.
29. Li J, Cheng K, Wang S, Morstatter F, Trevino RP, Tang J, Liu H. Feature selection: a data perspective. ACM Computing Surveys (CSUR). 2017;50(6):94.
30. Lichman M. UCI machine learning repository. 2013. http://archive.ics.uci.edu/ml.
31. Liu H, Motoda H. Feature selection for knowledge discovery and data mining, vol 454 Springer Science & Business Media. 2012.
32. Mafarja M, Aljarah I, Heidari AA, Faris H, Fournier-Viger P, Li X, Mirjalili S. Binary dragonfly optimization for feature selection using time-varying transfer functions. Knowl.-Based Syst. 2018;161:185–204. https://doi.org/10.1016/j.knosys.2018.08.003.
33. Mafarja M, Aljarah I, Heidari AA, Hammouri A, Faris H, Ala'M AZ, Mirjalili S. Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems. Knowl-Based Syst. 2017;145:25–45.
34. Mafarja M, Heidari AA, Faris H, Mirjalili S, Aljarah I. Dragonfly algorithm: theory, literature review, and application in feature selection, pp 47–67. Cham: Springer International Publishing; 2020.
35. Mafarja M, Jarrar R, Ahmad S, Abusnaina A. Feature selection using binary particle swarm optimization with time varying inertia weight strategies. In: The 2nd international conference on future networks & distributed systems , Amman, Jordan. ACM; 2018.
36. Mafarja M, Mirjalili S. Whale optimization approaches for wrapper feature selection. Appl Soft Comput. 2018;62:441–53.
37. Mafarja M, Sabar NR. Rank based binary particle swarm optimisation for feature selection in classification. In: Proceedings of the 2nd international conference on future networks and distributed systems, ICFNDS '18. New York: ACM; 2018. p. 19:1–6. https://doi.org/10.1145/3231053.3231072. http://doi.acm.org/10.1145/3231053.3231072.
38. Mafarja M, Eleyan D, Jaber I, Hammouri A, Mirjalili S. Binary dragonfly algorithm for feature selection. In: 2017 international conference on new trends in computing sciences (ICTCS). IEEE; 2017. p. 12–7.
39. Mafarja M, Mirjalili S. Hybrid whale optimization algorithm with simulated annealing for feature selection. Neurocomputing. 2017;260:302–12.
40. Mirjalili S. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Comput & Applic. 2016;27(4):1053–73.
41. Mirjalili S, Lewis A. S-shaped versus v-shaped transfer functions for binary particle swarm optimization. Swarm Evol Comput. 2013;9:1–14.
42. Mirjalili S, Lewis A. The whale optimization algorithm. Adv Eng Softw. 2016;95:51–67.
43. Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. Adv Eng Softw. 2014;69:46–61.
44. Mohamed F, AbdelNasser M, Mahmoud K, Kamel S. Accurate economic dispatch solution using hybrid whale-wolf optimization method. In: 2017 nineteenth international Middle East Power systems conference (MEPCON). IEEE; 2017. p. 922–7.
45. Molina D, LaTorre A, Herrera F. An insight into bio-inspired and evolutionary algorithms for global optimization: review, analysis, and lessons learnt over a decade of competitions. Cogn Comput. 2018;10:1–28.
46. Moradi P, Gholampour M. A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy. Appl Soft Comput. 2016;43:117–30.
47. Nguyen HB, Xue B, Liu I, Zhang M. Filter based backward elimination in wrapper based pso for feature selection in classification. In: 2014 IEEE congress on evolutionary computation (CEC). IEEE; 2014. p. 3111–8.
48. Oreski S, Oreski G. Genetic algorithm-based heuristic for feature selection in credit risk assessment. Expert systems with Applications. 2014;41(4):2052–64.
49. Pang J, Zhao Y, Xu J, Gu Y, Yu G. Super-graph classification based on composite subgraph features and extreme learning machine. Cogn Comput. 2018;10:1–15.
50. Saxena A, Soni BP, Kumar R, Gupta V. Intelligent Grey Wolf Optimizer–Development and application for strategic bidding in uniform price spot energy market. Appl Soft Comput. 2018;69:1–13.
51. Saxena A. A comprehensive study of chaos embedded bridging mechanisms and crossover operators for grasshopper optimisation algorithm. Expert Systems with Applications. 2019;132: 166–188.
52. Saxena A, Kumar R, Das S. β-chaotic map enabled grey wolf optimizer. Appl Soft Comput. 2019;75:84–105.
53. Singh N, Hachimi H. A new hybrid whale optimizer algorithm with mean strategy of grey wolf optimizer for global optimization. Mathematical and Computational Applications. 2018;23(1): 14.
54. Storn R, Price K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim. 1997;11(4):341–59.
55. Talbi EG. Metaheuristics: from design to implementation, vol 74 John Wiley & Sons. 2009.
56. Taradeh M, Mafarja M, Heidari AA, Faris H, Aljarah I, Mirjalili S, Fujita H. An evolutionary gravitational search-based feature selection. Inf Sci. 2019;497:219–39. https://doi.org/10.1016/j.ins.2019.05.038.
57. Wolpert DH, Macready WG, et al. No free lunch theorems for optimization. IEEE Trans Evol Comput. 1997;1(1):67–82.
58. Wootton AJ, Taylor SL, Day CR, Haycock PW. Optimizing echo state networks for static pattern recognition. Cogn Comput. 2017;9(3):391–399.

59. Xue B, Zhang M, Browne WN. Novel initialisation and updating mechanisms in pso for feature selection in classification. In: European conference on the applications of evolutionary computation. Springer; 2013. p. 428–438.
60. Yang XS, Deb S, Mishra SK. Multi-species cuckoo search algorithm for global optimization. Cogn Comput. 2018;10:1–11.
61. Zawbaa HM, Emary E, Parv B. Feature selection based on antlion optimization algorithm. In: 2015 third world conference on complex systems (WCCS). IEEE; 2015. p. 1–7.
62. Zorarpacı E, Özel SA. A hybrid approach of differential evolution and artificial bee colony for feature selection. Expert Syst Appl. 2016;62:91–103.

Affiliations

Majdi Mafarja[1] · Asma Qasem[2] · Ali Asghar Heidari[3,4] · Ibrahim Aljarah[5] · Hossam Faris[5] · Seyedali Mirjalili[6]

Majdi Mafarja
mmafarja@birzeit.edu

Asma Qasem
adaghra@gmail.com

Ali Asghar Heidari
as_heidari@ut.ac.ir; aliasghar68@gmail.com;
aliasgha@comp.nus.edu.sg; t0917038@u.nus.edu

Hossam Faris
hossam.faris@ju.edu.jo;
http://www.evo-ml.com

Seyedali Mirjalili
seyedali.mirjalili@griffithuni.edu.au;
http://www.alimirjalili.com

[1] Department of Computer Science, Birzeit University, Birzeit, Palestine

[2] Faculty of Engineering and Technology, Birzeit University, PoBox 14, Birzeit, Palestine

[3] School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran, Tehran, Iran

[4] Department of Computer Science, School of Computing, National University of Singapore, Singapore, Singapore

[5] Business Information Technology Department, King Abdullah II School for Information Technology, The University of Jordan, Amman, Jordan

[6] School of Information and Communication Technology, Griffith University, Nathan, Brisbane, QLD 4111, Australia