IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 16, 2019, accepted January 30, 2019, date of publication February 18, 2019, date of current version March 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2897325

A New Hybrid Algorithm Based on Grey Wolf Optimization and Crow Search Algorithm for Unconstrained Function Optimization and Feature Selection

SANKALAP ARORA[1], HARPREET SINGH[1], MANIK SHARMA[2], SANJEEV SHARMA[2], AND PRIYANKA ANAND[3]

[1]Department of Computer Science and Engineering, DAV University, Jalandhar 144012, India
[2]Department of Computer Science and Applications, DAV University, Jalandhar 144012, India
[3]Department of Computer Science and Engineering, Lovely Professional University, Jalandhar 144411, India

Corresponding author: Sankalap Arora (sankalap.arora@gmail.com)

ABSTRACT Grey wolf optimizer (GWO) is a very efficient metaheuristic inspired by the hierarchy of the Canis lupus wolves. It has been extensively employed to a variety of practical applications. Crow search algorithm (CSA) is a recently proposed metaheuristic algorithm, which mimics the intellectual conduct of crows. In this paper, a hybrid GWO with CSA, namely GWOCSA is proposed, which combines the strengths of both the algorithms effectively with the aim to generate promising candidate solutions in order to achieve global optima efficiently. In order to validate the competence of the proposed hybrid GWOCSA, a widely utilized set of 23 benchmark test functions having a wide range of dimensions and varied complexities is used in this paper. The results obtained by the proposed algorithm are compared to 10 other algorithms in this paper for verification. The statistical results demonstrate that the GWOCSA outperforms other algorithms, including the recent variants of GWO called, enhanced grey wolf optimizer (EGWO) and augmented grey wolf optimizer (AGWO) in terms of high local optima avoidance ability and fast convergence speed. Furthermore, in order to demonstrate the applicability of the proposed algorithm at solving complex real-world problems, the GWOCSA is also employed to solve the feature selection problem as well. The GWOCSA as a feature selection approach is tested on 21 widely employed data sets acquired from the University of California at Irvine repository. The experimental results are compared to the state-of-the-art feature selection techniques, including the native GWO, the EGWO, and the AGWO. The results reveal that the GWOCSA has comprehensive superiority in solving the feature selection problem, which proves the capability of the proposed algorithm in solving real-world complex problems.

INDEX TERMS Grey wolf optimizer, crow search algorithm, hybrid algorithm, function optimization, feature selection.

I. INTRODUCTION

Optimization is a process of searching the most optimal solution among all the available solutions of a particular problem. In consideration of the nature of optimization algorithms, these algorithms can be categorized broadly into two groups, i.e., deterministic algorithms and stochastic intelligent algorithms [1], [2]. In the case of deterministic algorithms, identical solutions are produced if its initial starting values are the same with each other when solving the same problem. In contradiction to deterministic algorithms, stochastic algorithms utilize random steps in order to reach the optima. In this, the optimization process cannot be repeated under any conditions. However, same final optimal solutions can be achieved by both of them in most of the cases. Stochastic algorithms are further classified into two types, i.e., heuristic algorithms and metaheuristic algorithms [3]. Heuristic, as the name suggests, is the process of finding the solutions by trial and error method whereas metaheuristic algorithms solve the optimization problems

The associate editor coordinating the review of this manuscript and approving it for publication was Huaqing Li.

stochastically having some prior knowledge about the random search [4]. It is a process of optimization which starts with a random solution, then explores and exploits the search space randomly with a specific probability. Since the last two decades, nature-inspired metaheuristic algorithms have become very popular due to their powerful and efficient performance in dealing with high-dimensional nonlinear optimization problems [3], [5]. These algorithms have the ability to exploit the useful information of the population in order to find the optimal solutions. Until now, substantial research has been done by various researchers on these algorithms and several nature-inspired metaheuristic algorithms have been introduced such as Particle Swarm Optimization (PSO) [6], Genetic algorithm (GA) [7], Differential Evolution (DE) [8], Bat Algorithm (BA) [9], Firefly Algorithm (FA) [10], [11], Butterfly Optimization Algorithm (BOA) [12], Grey Wolf Optimization (GWO) algorithm [13] and Crow Search algorithm (CSA) [14].

A major application of metaheuristic algorithms is in the domain of feature selection which deals with the dimensions of the data set in order to make predictions, however, when the dimensionality of the data sets is increased then the performance of the classification methods is considerably affected [15]. Moreover, high dimensional data sets have various disadvantages such as large time for model construction, redundant data and degraded performance which makes data analysis very difficult. To resolve this issue, feature selection is utilized as a major preprocessing step with the goal of selecting a subset of features out of the large data set as well as increase the accuracy of the classification or clustering model leading to the removal of noisy, extraneous and ambiguous data. In the past, various attempts have been made to employ metaheuristic algorithms such as PSO [16], [17], GA [18], [19] and record-to-record travel algorithm [20], [21] have been employed to solve feature selection problems. Furthermore, some recent algorithms including GWO [22], ant lion optimizer [23], flower pollination algorithm [24] and whale optimization based approaches [25] have also been used to find the optimal feature subset. Recently, binary butterfly optimization based approaches have been employed to solve the feature selection problem [26].

While designing or employing a metaheuristic, it must be kept in the mind that diversification (exploration of the search space) and intensification (exploitation of the optimal solutions obtained so far) must be balanced in an efficient manner. In this regard, one significant alternative is to develop a memetic algorithm in which (at least) two algorithms are integrated together with the aim to enhance the overall performance. The aim of the current study is to propose a hybrid algorithm focussing on the integration of GWO and CSA to demonstrate superior performance on global optimization and general classification problems.

GWO in fact is a metaheuristic algorithm, inspired by the leadership behavior and unique mechanism of hunting of grey wolves. This population-based metaheuristic has the ability to avoid local optima stagnation to some extent [3]. It also has good convergence ability towards the optima. In general, GWO advances itself strongly to exploitation. However, it cannot always implement exploration well. Thus, in some cases, GWO cannot always deal with the problem successfully and fails to find the global optimal solution. CSA is a recently proposed metaheuristic algorithm which mimics the intelligent behavior of crows. The authors utilized the intelligence of crows shown in a group in order to communicate with each other, hide and retrieve food. The strength of CSA lies in the ability to avoid local optima easily when dealing with complex, high dimensional and multimodal problems. On the contrary, the local search strategy of CSA is not very much efficient.

Considering the strengths of GWO and CSA, these two algorithms are ideal for hybridization. Therefore, in this study, a hybrid algorithm comprising GWO and CSA, termed GWOCSA is proposed which combines the two algorithms in order to achieve a more suitable trade-off between diversification and intensification, and offer significantly better results than the conventional GWO and CSA in terms of solution accuracy and convergence speed.

The major contributions of this research work are summarized as follows:

1) A novel hybridization approach based on GWO and CSA is proposed.
2) The proposed approach is applied on 23 function optimization benchmark problems.
3) The proposed hybrid approach is employed to solve feature selection problem and the results are validated on 21 data sets.
4) The performance of the proposed approach is compared with conventional GWO, CSA and eight other algorithms; Bat Algorithm (BA), Biogeography-based optimization (BBO), Dragonfly Algorithm (DA), Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Satin Bird Optimizer (SBO), Enhanced GWO (EGWO) and Augmented GWO (AGWO).

The remainder of the paper is arranged as follows: Section II presents the related works. Section III presents the background information of GWO and CSA focussing on their inspiration and mathematical model. The proposed hybrid algorithm is presented in Section IV whereas the experimental results on function optimization problems as well as feature selection problems are discussed in Section V. Finally, conclusions and future work are stated in Section VI.

II. RELATED WORKS

In the last few decades, hybrid algorithms have been utilized by various researchers to solve a variety of problems in the optimization field [27]. These hybrid algorithms have shown better performance in comparison to their counterparts in solving various complex problems [28]. In [29], GA has been hybridized with PSO for global optimization. In this research, the authors have utilized the hybridization of GA and PSO to generate individuals not only from the crossover and mutation operators but also by global and local search operators of PSO. GA has also been hybridized with Taguchi method for global numerical optimization [30]. In this hybrid algorithm, the Taguchi method is appended as an additional step in GA in order to choose higher quality genes to attain superior performance. In [31], grey wolf optimizer is hybridized with hybrid differential evolution algorithm for solving continuous global optimization problems. Nabil [32] introduced a hybrid algorithm between flower pollination algorithm and clonal selection algorithm to find more accurate solutions than the conventional FPA. The performance of the proposed hybrid algorithm is proved using various benchmark test problems. Tawhid and Ali [33] proposed a novel approach which hybridized GWO and GA, and employed the proposed approach to minimize the energy function of a simplified model of the molecule. Jayabarathi *et al.* [34] hybridized GWO with crossover and mutation operators of GA to achieve improved performance in solving economic dispatch problems. In another study, the exploration capability of GWO is hybridized with the ability of exploitation in PSO in order to improve the performance [35]. Recently, in order to enhance the performance of complex systems, GWO is hybridized with artificial bee colony and employed to optimize the parameters [36]. This research work focussed on utilizing the bee's information sharing strategy of artificial bee colony algorithm to perform exploration in addition to the exploitation capability of GWO. In [37], BBO and GWO are hybridized together to balance exploration and exploitation and to obtain superior performance than BBO and GWO individually. Hassanien *et al.* [38] hybridized CSA with a rough searching scheme in order to handle the impreciseness and roughness of the existing information regarding the global optimal solution and ultimately improving the performance of CSA. These studies confirm that the hybrid algorithms demonstrate superior performance in comparison to local or global search algorithms.

In the feature selection domain, various hybrid algorithms have been proposed with a great deal of success. The first hybrid metaheuristic algorithm for feature selection was proposed in 2004 which focussed on hybridizing local search techniques in GA to control the search process [39]. Another hybrid algorithm was proposed for feature selection in [40] which focussed on hybridization of GA and PSO algorithms with SVM classifier and employed to microarray data classification. Moreover in [41], PSO is hybridized with a local search algorithm to guide the search process in an efficient manner in order to select the minimal reducts depending on their correlation data. Hybrid algorithms based on ant colony optimization using GA [42], [43] and cuckoo search [44] are proposed in the same field.

Moreover, hybrid algorithms have also been utilized for feature selection such as hybridization of differential evolution and artificial bee colony has been done [45] whereas hybrid algorithms based on the whale optimization algorithm and simulated annealing are also employed for feature selection [46]. Recently, an approach based on grasshopper optimization algorithm [47] and evolutionary population dynamics [48] has been successfully employed in [49] as a solution for selecting the optimal feature set. Interested readers are referred to [50] and [51] for further reading about metaheuristics and feature selection.

Despite the advantages of the aforementioned hybrid algorithms for function optimization and feature selection, one might question the intention of developing a new hybrid algorithm. This question can be responded using the No-Free-Lunch (NFL) which rationally proves that any single algorithm is not capable of solving all optimization problems [52]. This means that there is always room for developing new algorithms to solve the function optimization problems as well as feature selection problems in a more efficient manner. This inspired our efforts to propose yet another hybrid algorithm for function optimization and feature selection problems.

III. METHODS

A. GREY WOLF OPTIMIZER

Grey Wolf Optimizer (GWO) was proposed by Mirjalili *et al.* [13] in 2014. It is a newcomer in the field of nature-inspired metaheuristic algorithms. It mimics the leadership and hunting characteristics of the grey wolves. Grey wolves are members of Canidae family and they follow a very strict social hierarchy. They prefer to hunt for prey in a pack of 5-12 wolves. Some assumptions have been taken in the conventional GWO for its efficient simulation which includes the four levels in wolves' hierarchy that are alpha (α), beta (β), delta (δ) and omega (ω). α wolf is at the topmost level being the leader of the wolf pack. It can be a male or a female wolf. It is responsible for taking all types of decisions like hunting, maintaining discipline, sleeping and waking time for whole pack [53]. The second level is β which are the subordinate wolves and help the α leader in decision making or other activities. β wolf being the second best in the group has the highest probability to become α leader in the group. The third level of grey wolves, i.e., δ wolves, dominate the wolves of forth and the last level called the ω wolves which are responsible for maintaining safety and integrity in the wolf pack [13]. GWO is mathematically modeled into four phases which are described as follows:

1) HIERARCHICAL STRUCTURE

The GWO algorithm is mathematically modeled on the basis of the social hierarchy of the wolves. The top level of the social hierarchy, i.e., α is considered as the best solution found. Similarly, β and δ are considered as second and third best solution respectively. ω wolves are assumed to be the remaining candidate solutions that follow α, β and δ wolves [13].

2) ENCIRCLING PREY

All the grey wolves have some inherent characteristic of encircling the prey during hunting. The mathematical equations modeled in GWO for encircling characteristic of the wolves are represented in Eq. (1) and Eq. (2) [13].

$$\overrightarrow{D} = |\overrightarrow{C}.\overrightarrow{X_p}(t) - \overrightarrow{X}(t)| \tag{1}$$

$$\overrightarrow{X}(t+1) = \overrightarrow{X_p}(t) - \overrightarrow{A}.\overrightarrow{D} \tag{2}$$

Here, $\vec{D}$ is the distance from the prey to the wolf. $\vec{X}$ represents the position vector of the wolf and $\vec{X_p}$ indicates the position vector of the prey at iteration t. $\vec{A}$ and $\vec{C}$ are the vectors which are random and are calculated as shown in Eq. (3) and Eq. (4) [13].

$$\overrightarrow{A} = 2\overrightarrow{a}.\overrightarrow{r_1} - \overrightarrow{a} \tag{3}$$

$$\overrightarrow{C} = 2.\overrightarrow{r_2} \tag{4}$$

Here, $\vec{r_1}$ and $\vec{r_2}$ are the random vectors in the range of [0, 1]. These vectors make wolves to reach at any point between the prey and the wolf. Vector $\vec{a}$ is involved in controlling activity of the GWO algorithm and used in calculating $\vec{A}$. The component values of $\vec{a}$ vector decreases linearly from 2 to 0 over the courses of iterations [13].

3) HUNTING THE PREY

Having the ability to recognize the location of the prey, the grey wolves can easily encircle it. The α wolf guides the whole hunting process. All the grey wolves do hunting according to α, β and δ wolves. They also update their positions according to the resultant best position of α, β and δ wolves. Mathematically, it is formulated in Eq. (5), Eq. (6) and Eq. (7) [13].

$$\overrightarrow{D_\alpha} = |\overrightarrow{C_1}.\overrightarrow{X_\alpha} - \overrightarrow{X}|, \overrightarrow{D_\beta} = |\overrightarrow{C_2}.\overrightarrow{X_\beta} - \overrightarrow{X}|, \overrightarrow{D_\delta} = |\overrightarrow{C_3}.\overrightarrow{X_\delta} - \overrightarrow{X}| \tag{5}$$

$$\overrightarrow{X_1} = \overrightarrow{X_\alpha} - \overrightarrow{A_1}.\overrightarrow{D_\alpha}, \overrightarrow{X_2} = \overrightarrow{X_\beta} - \overrightarrow{A_2}.\overrightarrow{D_\beta}, \overrightarrow{X_3} = \overrightarrow{X_\delta} - \overrightarrow{A_3}.\overrightarrow{D_\delta} \tag{6}$$

The updated position of the grey wolf can be calculated using Eq. (7).

$$\overrightarrow{X}(t+1) = (\overrightarrow{X_1} + \overrightarrow{X_2} + \overrightarrow{X_3})/3 \tag{7}$$

The pseudo code of the GWO algorithm is presented in Fig. 1.

4) SEARCHING AND ATTACKING THE PREY

Grey wolves attack the prey only when it stops moving. Mathematically, it is modelled on the basis of $\vec{a}$ vector used in Eq. (3). $\vec{A}$ is a random vector whose value lies in the range $[-a, a]$, where the value of $\vec{a}$ is decreased from 2 to 0 over the course of iterations using Eq. (8).

$$\overrightarrow{a} = 2 - (2 \times t/Max_{iter}) \tag{8}$$

So, if $|\vec{A}| < 1$, it means the wolf will be forced to attack the prey by going towards it and if $|\vec{A}| > 1$, the wolf will get diverge out from the prey and will search out for a fitter prey [13]. The process of searching for prey by grey wolves is done according to the location of the α, β and δ wolves. The exploitation and exploration depend only on the values of $\vec{A}$

Algorithm 1 Pseudo Code of GWO [13]

Initialize *the population of grey wolves* $X_i (i = 1, 2, \ldots, n)$
Initialize *parameters a, A and C*
Calculate *the fitness of each Search_agent (Wolf)*
X_α *= The best search_agent/wolf*
X_β *= The second best search_agent/wolf*
X_δ *= The third best search_agent/wolf*
while *(t<Max_iterations)*
  for *each search_agent*
    Update *the position of current search_agent using Eq.(7),*
  end for
  Update *parameters a, A, C*
  Calculate *the fitness of all search agents.*
  Update $X_\alpha, X_\beta, X_\delta$
t=t+1
end while
return X_α

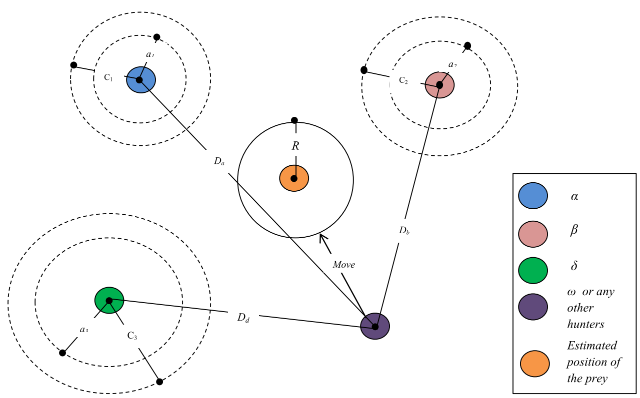

FIGURE 1. Position updating in GWO [13].

and $\vec{C}$ vector. We can utilize the random values of $\vec{A}$ to make the wolf diverge or converge from the prey. Random values of $\vec{C}$ vector lie in the range [0, 2] which plays an important role in avoiding local optima stagnation. It adds some random weight to the prey to make it more difficult for grey wolves to define the distance from the prey to the wolf. $\vec{C} > 1$ means $\vec{C}$ is emphasizing the effect of prey and if $\vec{C} < 1$, $\vec{C}$'s effect will get de-emphasize stochastically. In the whole process, mainly $\vec{A}$ and $\vec{C}$ vectors are to be adjusted carefully. Both the parameters will emphasize or deemphasize exploitation or exploration. In the end, when the last criterion is satisfied, the GWO algorithm will get terminated and the best position of the alpha wolf will be the outcome. The diagrammatical representation of the effect of parameters $\vec{A}$ and $\vec{C}$ in updating the wolves' position is represented in Algorithm 1.

B. CROW SEARCH ALGORITHM

Askarzadeh proposed a nature-inspired algorithm which mimics the food hiding mechanism of crows in 2016 [14]. Basically, a crow is considered an intelligent bird that has a sharp brain and it possess the ability which is utilized to warn its species about some dangerous situations. One of the best aspects of their intelligence is that they can effectively hide and easily remember its food location. The working of CSA is depended on four major principles, i.e., they live in a flock, remember the location of the hidden food, follow another member of their species and finally, shield their caches from being pilfered stochastically. Due to its simplicity and efficiency, CSA has been used to solve different problems related to the selection of conductor size [54], feature selection [55], [56], image segmentation [57] and electromagnetic optimization [58].

C. MATHEMATICAL MODEL OF CSA

The working principle of CSA is based upon the ability of crows in hiding and retrieving the food. The complete working of the CSA is presented below:

1) In the first step, the optimization problem along with its decision variables and constraints is formulated and the values of adjustable parameters, i.e., flock size (N, the maximum number of iterations($Max_{iterations}$), Flight Length (fl) and Awareness Probability (AP) are set.
2) A matrix of N rows and d columns is designed where N, d is a number of crows and decision variables respectively. Each crow represents a feasible solution. Here, the memory for each crow is setup. Moreover, in the beginning, all crows do not have any experience so it is assumed that each crow has concealed their food at random locations.
3) Now, the value of fitness function is computed.
4) In this step, the new position of the crow is computed. First, a random number is generated and compared to awareness probability and if the value of the random number is smaller than awareness probability then the crow moves randomly in the search space. Otherwise, the crow (x^i) randomly selects any flock crow (m^j) and then follows m^j to determine the location of its hidden food. The new position for the crow (x^i) is calculated as:
$$x^{i+1,t+1} = x^{i,t} + r_i \times fl^{i,t} \times (m^{j,t} - x^{i,t}) \tag{9}$$
Here r represents a random number whereas the iteration count is demonstrated by t. This process will be repeated for all N crows.
5) The fitness of the new position will be computed.
6) Based upon the fitness value of new and memorized position, i.e., if the fitness value of the newly occupied position is better than the memorized one, then the memory of the crow will be updated.
7) The steps from 4 to 6 are repeated until maximum iterations or the termination criterion is met. The best position of the crow memory will represent the final solution of the optimization problem.

The pseudo code of the CSA algorithm is presented in Algorithm 2.

Algorithm 2 Pseudo Code of CSA [14]

```
Initialize the population of crows X_i(i = 1, 2, ..., n)
Calculate the fitness of each crow
Initialize the memory of crows
while (t<Max_iterations)
  for each crow
    Define an Awareness Probability (AP)
    Generate a random number r
    if r ≥ AP
      Update the position of crow using Eq.(9),
    else
      Generate the position of crow randomly
    endif
  end for
  Check the feasibility of new positions
  Calculate the fitness of all search agents.
  Update the memory of crows
  t=t+1
end while
return the best crow
```

IV. THE PROPOSED APPROACH

A. MODIFIED POSITION UPDATION MECHANISM

It is a well established fact that both exploration, as well as exploitation, is essential for any population-based algorithm to demonstrate excellent performance. In classical GWO, the foremost matter of concern is that all the search agents (wolves) are updated according to the α (best search agent), β (second best search agent) and δ (third best search agent) in the whole optimization process as shown in Eq. (7). Basically, this position updating mechanism leads to premature convergence because the search agents were not allowed to explore the search space efficiently. Moreover, the same optimization process as mentioned in Eq. (7) provides limited exploitation capability in the later stages of optimization which leads to slow convergence.

Therefore in order to overcome the aforementioned limitations of the conventional GWO, it is hybridized with CSA to achieve a more suitable balance between exploitation and exploration. Specifically, CSA incorporates a control parameter fl in its position updation equation as mentioned in Eq. (9) which allows the search agents to decide the magnitude of the step movement towards the other search agent. This parameter plays a very significant role in the attaining the global optima as the large value of fl lead of global exploration while a small value of fl results to local exploitation. Fig. 2 demonstrates the effect of fl in the searching process of CSA. As already mentioned earlier, GWO has good exploitation ability but poor exploration capability, therefore in the proposed GWOCSA, a larger value of fl is utilized in order to utilize the CSA's excellent exploration quality as shown in Eq. (10). This means, the proposed algorithm can effectively maximize the two algorithms' advantages and therefore, it can obtain strong universal applicability. In GWOCSA, instead of updating from α, β and δ, a search

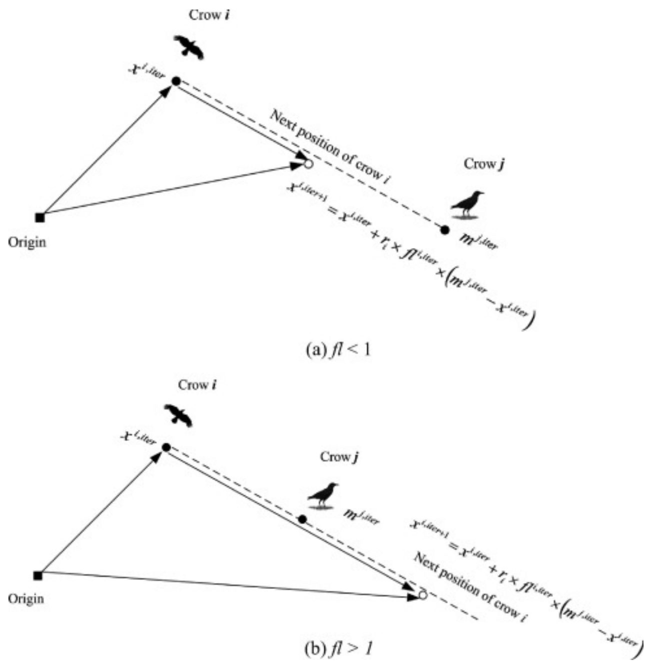

FIGURE 2. Position updating in CSA [14].

agent is allowed to update its position only using α and β as shown in Eq. (10).

$$\vec{X}(t+1) = \vec{X} + f1 \times rand \times ((\vec{X_1} - \vec{X}) + (\vec{X_2} - \vec{X}))/2 \tag{10}$$

Another signification addition, in order to maintain population diversity, not all individuals in the population are updated by the alpha and beta updating direction, but by alpha only in the proposed GWOCSA. This acts as a shrinking strategy which enables the proposed algorithm to escape from local optimum.

$$\vec{X}(t+1) = \vec{X} + f1 \times rand \times (\vec{X_1} - \vec{X}) \tag{11}$$

B. ADAPTIVE BALANCE PROBABILITY (p) STRATEGY

Although, the proposed GWOCSA possess the excellent capabilities of exploration and exploitation of CSA and GWO, however, a proper balance of these two phases must be achieved in order to achieve good performance. In an ideal scenario, an algorithm must attain the ability to explore a huge search space in the early optimization stage to avoid premature convergence while exploiting small regions in the later optimization phases to efficiently refine the final solutions. This means, in order to attain the required exploration-exploitation ratio, a fixed balance probability between Eq. (10) and Eq. (11) is not favorable. Therefore, in this study, an adaptive balance probability is proposed which allows the GWOCSA to achieve acceleration throughout early steps of optimization process whereas in the later stages of optimization promising solutions will possess a high probability to be exploited. The adaptive balance probability

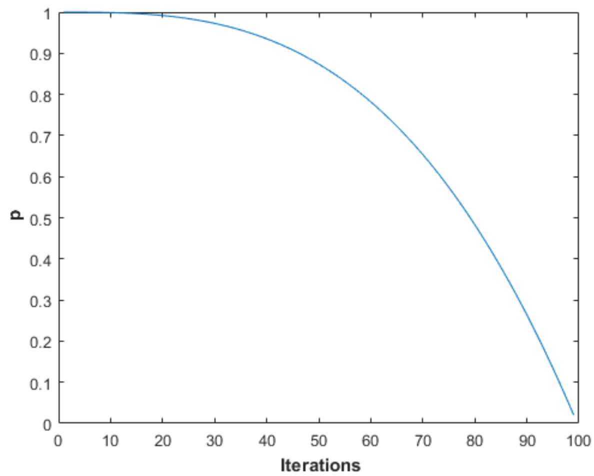

FIGURE 3. Adaptive balance probability *p*.

(p) is computed as follows:

$$p = 1 - (1.01 \times t^3 / Max\_iter^3) \tag{12}$$

where t denotes current iteration and Max_iter denotes maximum number of iterations.

C. NONLINEAR CONTROL PARAMETER (α) STRATEGY

As mentioned in the previous section that the parameter A plays a very significant role in balancing the exploration and exploitation of a search agent. Specifically, the parameter A is critically dependent on $\vec{a}$ which ultimately controls the direction of the search process. A larger value of $\vec{a}$ facilitates exploration phase whereas a smaller value facilitates exploitation. This means a suitable selection of $\vec{a}$ can offer an upright balance of exploration and exploitation which can lead to superior performance. In the classical GWO, the value of $\vec{a}$ is linearly decreased from 2 to 0 using Eq. (3). Till now, several mechanism of updating control parameter $\vec{a}$ have been proposed, such as [59] and [60]. Therefore, it can be observed that superior performance can be achieved if the values of control parameter $\vec{a}$ are selected by using a nonlinearly decreasing approach, instead of a linearly decreasing approach. Using the above information, an improved strategy, as shown in Eq. (13), is utilized to generate the values for control parameter $\vec{a}$ during the optimization process. This strategy allows the proposed algorithm to effectively explore the search space in comparison to traditional GWO.

$$a = 2 - (cos(rand()) \times 1/Max\_iter) \tag{13}$$

The pseudo code of the proposed GWOCSA algorithm is shown in Algorithm 3.

V. EXPERIMENTS AND RESULTS

In order to extensively investigate the performance of GWOCSA algorithm, the experiments were conducted in two sets. In the first set of experiments, a benchmark function set containing 23 test problems having different

Algorithm 3 Pseudo Code of the Proposed GWOCSA

Initialize *the population of grey wolves* $X_i (i = 1, 2, \ldots, n)$
Initialize *parameters a, A and C*
Calculate *the fitness of each Search_agent (Wolf)*
X_α *= The best search_agent/wolf*
X_β *= The second best search_agent/wolf*
while *(t<Max_iterations)*
  for *each search_agent*
    if *p>rand*
      Update *the position of current search_agent using Eq.(10)*
      else
      Update *the position of current search_agent using Eq.(11)*
    endif
  end for
Update *the value of p using Eq. (12)*
Update *the value of a using Eq. (13)*
Update *parameters A, C*
Calculate *the fitness of all search agents.*
Update X_α, X_β
t=t+1
end while
return X_α

characteristics is employed whereas in the second set of experiments twenty-one data sets from UCI repository are utilized in order to compare the performance of GWOCSA with other metaheuristics as feature selection approaches. The algorithms which are utilized in the comparative study are BA [9], BBO [61], CSA [14], DA [62], GA [63], GWO [13], PSO [64], SBO [65], EGWO [66] and AGWO [67]. Table 1 shows the initial values of critical parameters for the algorithms used in this study which are selected according to the literature. The maximum limit on number of iterations is set to 300 for benchmark functions whereas for feature selection problem, it is set to 100 [68]. Lastly, the size of the population is fixed to 30 for function optimization problem and 7 for feature selection problem. All the results are reported on the average of 30 independent runs in order to achieve statistically meaningful results. All the algorithms are implemented using MATLAB R2009b, under Microsoft Windows 8 operating system. All simulations are carried out on a CPU $i5 - 3210$ (Intel CoreTM Processor @2.50 GHz) computer.

A. FUNCTION OPTIMIZATION EXPERIMENTS

In order to evaluate the performance of the proposed GWOCSA, various experiments on a diverse subset of function optimization problems are done in this section. These functions have been widely employed by different researchers in order to validate the performance of optimization algorithms [69], [70]. These functions are divided into three categories on the basis of dimensionality and modality. In the first category, unimodal functions (F1-F7) investigate

TABLE 1. Parameter settings of the algorithms used for comparison in the current study.

	Parameter	Value (s)
BA	Pulse rate	0.5
	Loudness	0.5
	Frequency	Min=0 and Max=1
BBO	Immigration probability	[0,1]
	Mutation probability	0.05
	Habitat modification probability	1.0
	Step size	1.0
	Migration rate	1.0
	Maximum immigration	1.0
CSA	Flight length	2
	Awareness probability	0.1
DA	Inertia weight	0.9-0.2
	Separation weight	0.1
	Alignment weight	0.1
	Cohesion weight	0.7
	Food attraction weight	1
	Enemy distraction weight	1
GA	Mutation ratio in GA	0.1
	Crossover ratio in GA	0.9
	Selection mechanism in GA	Roulette wheel
GWO	*a*	linearly decreased from 2 to 0
LSA	Channel time	10
PSO	Inertia w in PSO	[0.9, 0.6]
	Acceleration constants in PSO	[2, 2]
SBO	Step size	0.94
	Mutation probability	0.05
	Difference between the upper and lower limit	0.02

the convergence speed of an algorithm which are shown in Table 2. In the second category, multimodal functions (F8-F13) assess the ability of an algorithm to find global optima when the number of local optima increases exponentially with (tuned) problem dimension as shown in Table 3. In the last category, i.e., fixed dimensional multimodal functions, the number of design variables cannot be altered and moreover, these functions offer dissimilar search space in comparison to multimodal benchmark functions. These fixed dimensional multimodal functions are shown in Table 4.

Tables 5- 7 show the results of all the algorithms on different benchmark functions. In each table, mean and standard deviation of the best solution obtained by every algorithm are reported and the best results are highlighted in bold. It can be observed that the performance of the proposed GWOCSA is superior to other algorithms in most of the benchmark functions. Table 5 demonstrates the comparison between the proposed GWOCSA with other algorithms for unimodal functions. It can be analyzed that GWOCSA has shown superior results by obtaining best mean values for six out of seven benchmark functions. As mentioned earlier, unimodal functions are utilized to assess the convergence towards the (only) optima. Therefore, the results reported in Table 5 prove that the hybridization of GWO and CSA has demonstrated better performance in comparison to GWO, CSA as well as other algorithms employed in this study. Additionally, in order to better validate the superior convergence behavior of GWOCSA on test functions, Fig. 4 is provided.

TABLE 2. Unimodal benchmark functions.

Formula	Dim	Range	f_{min}
F1(x)= $\sum_{i=1}^{n} x_i^2$	30	[-100,100]	0
F2(x)= $\sum_{i=1}^{n} \lvert x_i\rvert - \prod_{i=1}^{n} \lvert x_i\rvert$	30	[-10, 10]	0
F3(x)= $\sum_{i=1}^{n}\left(\sum_{j-1}^{i} x_j\right)^2$	30	[-100,100]	0
F4(x)= $\max(\lvert x_i\rvert, 1 \le i \le n)$	30	[-100,100]	0
F5(x)= $\sum_{i=1}^{n-1}[100(x_{i+1} - {x_i}^2)^2 + (x_i - 1)^2]$	30	[-30, 30]	0
F6(x)= $\sum_{i=1}^{n}([x_i + 0.5])^2$	30	[-100, 100]	0
F7(x)= $\sum_{i=1}^{n} i{x_i}^4 + \mathrm{rand}(0,1)$	30	[-1.28, 1.28]	0

TABLE 3. Multimodal benchmark functions.

Function	Dim	Range	f_{min}
F8(x)= $\sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$	30	[-500, 500]	−418.982
F9(x)=$\sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i) + 10)$	30	[-5.12, 5.12]	0
F10(x)= $-20\exp\left(-0.2 \times \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	30	[-32, 32]	0
F11(x)= $\frac{1}{4000}\sum_{i=1}^{n} {x_i}^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-600, 600]	0
F12(x)= $\frac{\pi}{n}\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_1 - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4}, \quad u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 - a & x_i < a \\ k(-x_i - a)^m x_i < -a \end{cases}$	30	[-50, 50]	0
F13(x)= $0.1\{\sin^2(3\pi x_i)\sum_{i=1}^{n}(x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	[-50, 50]	0

TABLE 4. Fixed-dimensional multimodal benchmark functions.

Function	Dim	Range	f_{min}
F14(x)= $\left(\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6}\right)^{-1}$	2	[-65, 65]	1
F15(x)=$\sum_{i=1}^{11}\left[a_i - \frac{x_i(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$	4	[-5, 5]	0.00030
F16(x)= $4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[-5, 5]	-1.0316
F17(x)= $\left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)cos x_1 - 10$	2	[-5, 5]	0.398
F18(x)= $[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 times(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[-2, 2]	3
F19(x)= $-\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	3	[1,3]	-3.86
F20(x)= $-\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	6	[0, 1]	-3.32
F21(x)= $-\sum_{i=1}^{5}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	-101532
F22(x)= $-\sum_{i=1}^{7}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.4028
F23(x)= $-\sum_{i=1}^{10}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.5363

Looking closely, it can be analyzed that GWOCSA tends to demonstrate a faster convergence rate in comparison to other algorithms in the first quarter of optimization. The underlying reason for the superior performance of the proposed algorithm is that it is able to locate high performing regions in the search space of the function optimization problem in hand.

TABLE 5. Results of unimodal benchmark functions.

F		BA	BBO	CSA	DA	GA	GWO	PSO	SBO	EGWO	AGWO	GWOCSA
F1	Mean	1.07E+04	2.28E+01	7.25E+01	6.33E+00	9.87E+02	4.00E-15	1.67E-03	4.27E-01	1.74E-16	2.45E-24	**1.01E-28**
	Std Dev	2.50E+03	9.67E+01	2.75E+01	1.33E+01	3.81E+02	4.52E-15	5.63E-03	1.39E-01	3.69E-16	5.37E-24	1.26E-28
F2	Mean	5.00E+05	6.65E-01	5.29E+00	1.94E+00	7.17E-03	1.31E-09	3.20E-01	2.47E-01	5.16E-11	1.44E-15	**1.50E-17**
	Std Dev	2.23E+06	9.78E-02	1.05E+00	1.15E+00	2.69E-02	6.24E-10	3.66E-01	3.91E-02	8.54E-11	1.20E-15	1.25E-17
F3	Mean	3.65E+04	1.59E+04	6.69E+02	3.47E+02	6.25E+02	5.06E-02	3.92E+02	1.94E+03	1.27E-01	3.66E-02	**5.18E-04**
	Std Dev	1.67E+04	4.75E+03	2.70E+02	8.52E+02	3.36E+02	6.80E-02	2.49E+02	6.78E+02	2.83E-01	1.63E-01	1.07E-03
F4	Mean	4.68E+01	4.99E+01	8.69E+00	4.49E+00	6.66E+00	9.85E-04	4.32E+00	8.92E+00	1.55E+00	6.39E-06	**2.07E-07**
	Std Dev	8.12E+00	7.26E+00	1.86E+00	3.45E+00	3.34E+00	6.86E-04	1.65E+00	3.55E+00	3.95E+00	1.08E-05	3.00E-07
F5	Mean	8.36E+06	2.41E+02	1.77E+03	1.25E+03	5.63E+02	2.74E+01	8.07E+01	4.54E+02	2.80E+01	2.73E+01	**2.70E+01**
	Std Dev	4.88E+06	2.24E+02	9.42E+02	2.22E+03	4.94E+02	7.68E-01	5.64E+01	4.74E+02	9.83E-01	6.85E-01	5.00E-01
F6	Mean	1.16E+04	1.29E+00	6.58E+01	1.47E+01	1.20E+01	1.03E+00	**1.20E-03**	4.36E-01	3.19E+00	1.49E+00	1.23E+00
	Std Dev	2.91E+03	1.78E+00	2.75E+01	2.45E+01	7.16E+00	4.13E-01	1.89E-03	1.86E-01	5.08E-01	3.13E-01	2.62E-01
F7	Mean	3.71E+00	1.79E-01	8.15E-02	3.98E-02	3.55E-02	3.92E-03	3.88E-02	3.28E-01	1.19E-02	3.17E-03	**1.92E-03**
	Std Dev	1.84E+00	6.27E-02	3.62E-02	2.99E-02	2.82E-02	2.79E-03	1.81E-02	8.69E-02	5.87E-03	2.27E-03	9.88E-04

TABLE 6. Results of multimodal benchmark functions.

F		BA	BBO	CSA	DA	GA	GWO	PSO	SBO	EGWO	AGWO	GWOCSA
F8	Mean	**-2.84E+71**	-1.02E+04	-6.26E+03	-2.75E+03	-1.22E+03	-6.16E+03	-6.84E+03	-5.92E+03	-6.34E+03	-3.66E+03	-3.57E+03
	Std Dev	7.36E+71	5.40E+02	5.62E+02	3.10E+02	7.00E+02	1.03E+03	6.46E+02	1.18E+03	6.13E+02	2.92E+02	4.42E+02
F9	Mean	6.63E+01	7.58E+00	4.35E+01	3.01E+01	3.34E+00	8.21E+00	5.22E+01	4.75E+01	1.87E+02	1.47E+00	**1.19E+00**
	Std Dev	2.18E+01	1.98E+00	1.08E+01	1.32E+01	2.51E+00	5.65E+00	1.34E+01	1.13E+01	5.25E+01	6.56E+00	3.32E+00
F10	Mean	1.45E+01	3.31E-01	4.43E+00	3.15E+00	2.03E+00	1.30E-08	1.86E+00	1.52E+00	1.30E-01	2.77E-13	**1.37E-14**
	Std Dev	8.62E-01	8.31E-02	7.44E-01	1.46E+00	1.45E+00	1.02E-08	8.57E-01	1.95E+00	5.83E-01	3.06E-13	3.53E-15
F11	Mean	1.13E+02	6.14E+01	1.57E+00	6.91E-01	9.95E-01	8.81E-03	1.50E-02	6.34E-01	9.89E-03	**0.00E+00**	**0.00E+00**
	Std Dev	3.11E+01	1.90E+01	2.75E-01	3.77E-01	2.11E-01	1.06E-02	2.09E-02	1.52E-01	1.18E-02	0.00E+00	0.00E+00
F12	Mean	4.55E+06	6.45E+00	6.32E+00	1.51E+00	1.99E+00	7.49E-02	3.39E-01	4.44E+00	3.00E+00	1.06E-01	**4.92E-02**
	Std Dev	7.23E+06	1.90E+00	1.90E+00	1.51E+00	1.61E+00	5.13E-02	4.70E-01	2.59E+00	3.55E+00	3.36E-02	8.54E-03
F13	Mean	2.42E+07	1.04E-01	1.82E+01	1.06E+00	3.99E-01	9.19E-01	2.58E-01	**5.68E-02**	2.70E+00	1.12E+00	9.39E-01
	Std Dev	1.22E+07	1.01E-01	1.77E+01	1.31E+00	3.21E-01	2.43E-01	2.93E-01	2.97E-02	5.52E-01	2.04E-01	2.09E-01

TABLE 7. Results of fixed-dimensional multimodal benchmark functions.

F		BA	BBO	CSA	DA	GA	GWO	PSO	SBO	EGWO	AGWO	GWOCSA
F14	Mean	1.01E+01	9.83E+00	1.15E+00	1.35E+00	2.04E+00	3.80E+00	4.83E+00	9.20E+00	6.42E+00	2.28E+00	**9.98E-01**
	Std Dev	5.40E+00	7.68E+00	6.64E-01	6.66E-01	1.41E+00	3.23E+00	3.45E+00	6.26E+00	5.03E+00	2.22E+00	2.21E-05
F15	Mean	4.81E-03	6.20E-03	4.12E-04	3.70E-03	3.28E-02	1.52E-03	2.35E-03	2.90E-03	7.58E-03	5.58E-03	**3.38E-04**
	Std Dev	5.56E-03	7.24E-03	2.96E-04	5.92E-03	3.07E-02	4.44E-03	6.16E-03	2.83E-03	9.76E-03	8.76E-03	2.22E-05
F16	Mean	**-1.03E+00**	**-1.03E+00**	**-1.03E+00**	**-1.03E+00**	**-5.07E-01**	**-1.03E+00**	**-1.03E+00**	**-9.91E-01**	**-1.03E+00**	**-1.03E+00**	**-1.03E+00**
	Std Dev	2.64E-09	1.74E-04	3.13E-15	5.59E-07	3.95E-01	9.33E-08	1.76E-16	1.82E-01	2.26E-08	7.15E-06	5.57E-06
F17	Mean	**3.98E-01**	**3.98E-01**	**3.98E-01**	**3.98E-01**	6.05E-01	**3.98E-01**	**3.98E-01**	**3.98E-01**	**3.98E-01**	**3.98E-01**	**3.98E-01**
	Std Dev	1.33E-09	1.72E-04	1.77E-15	1.23E-09	7.36E-02	3.79E-06	0.00E+00	7.30E-06	3.02E-07	2.10E-04	2.93E-04
F18	Mean	5.70E+00	5.75E+00	**3.00E+00**	**3.00E+00**	2.87E+01	**3.00E+00**	**3.00E+00**	2.09E+01	**7.05E+00**	**3.00E+00**	**3.00E+00**
	Std Dev	8.31E+00	8.39E+00	3.09E-14	2.82E-10	3.98E+01	1.73E-04	1.91E-15	3.38E+01	9.89E+00	7.55E-05	1.97E-05
F19	Mean	**-3.86E+00**	**-3.86E+00**	**-3.86E+00**	**-3.86E+00**	-2.11E+00	**-3.86E+00**	**-3.86E+00**	**-3.86E+00**	**-3.86E+00**	**-3.86E+00**	**-3.86E+00**
	Std Dev	1.12E-07	1.88E-03	2.43E-13	2.34E-03	8.66E-01	2.77E-03	2.22E-15	2.50E-07	2.15E-03	2.92E-03	2.52E-03
F20	Mean	-3.28E+00	-3.28E+00	-3.28E+00	-3.22E+00	-1.01E+00	-3.25E+00	-3.27E+00	-3.28E+00	-3.24E+00	-3.17E+00	**-3.31E+00**
	Std Dev	5.82E-02	5.52E-02	6.18E-02	1.08E-01	5.47E-01	7.52E-02	5.98E-02	5.82E-02	7.11E-02	1.03E-01	5.58E-03
F21	Mean	-5.54E+00	-5.15E+00	-8.41E+00	-7.01E+00	-4.66E+00	**-9.14E+00**	-5.02E+00	-5.62E+00	-5.26E+00	-5.45E+00	-6.80E+00
	Std Dev	3.55E+00	3.43E+00	3.14E+00	3.04E+00	3.04E+00	2.49E+00	3.18E+00	3.81E+00	3.08E+00	1.81E+00	2.23E+00
F22	Mean	-5.39E+00	-6.02E+00	-1.00E+01	-7.64E+00	-7.41E+00	**-1.04E+01**	-7.25E+00	-8.16E+00	-7.56E+00	-6.89E+00	-8.76E+00
	Std Dev	3.46E+00	3.70E+00	1.71E+00	2.88E+00	3.47E+00	2.30E-03	3.64E+00	3.52E+00	3.61E+00	1.08E+00	6.47E-01
F23	Mean	-5.43E+00	-7.17E+00	-9.36E+00	-6.09E+00	-4.79E+00	**-9.72E+00**	-7.51E+00	-5.79E+00	-7.02E+00	-7.28E+00	-8.82E+00
	Std Dev	3.87E+00	3.50E+00	2.87E+00	2.82E+00	3.11E+00	2.50E+00	3.84E+00	3.98E+00	4.02E+00	1.09E+00	5.52E-01

The rest of benchmark functions are multimodal, i.e., they have many local solutions. As we briefly mentioned before, this type of functions validates the performance of new algorithms in terms of its ability to avoid local optima stagnation. Tables 6 and 7 demonstrate that the performance of the proposed GWOCSA is superior to other algorithms

TABLE 8. *p*-values obtained from the rank-sum test on different benchmark functions ($p > 0.05$ are underlined).

	BA	BBO	CSA	DA	GA	GWO	PSO	SBO	EGWO	AGWO
F1	8.86E-05	8.86E-05	8.86E-05	8.86E-05	8.86E-05	8.86E-05	8.86E-05	8.86E-05	8.86E-05	8.86E-05
F2	8.86E-05	8.86E-05	8.86E-05	1.03E-04	1.37E-02	8.86E-05	8.86E-05	8.86E-05	8.86E-05	8.86E-05
F3	8.86E-05	8.86E-05	8.86E-05	8.86E-05	8.86E-05	8.86E-05	8.86E-05	8.86E-05	1.20E-04	3.70E-01
F4	8.86E-05	8.86E-05	8.86E-05	8.86E-05	8.86E-05	8.86E-05	8.86E-05	8.86E-05	8.86E-05	1.89E-04
F5	8.86E-05	8.86E-05	8.86E-05	7.80E-04	1.03E-04	5.22E-02	2.54E-04	8.86E-05	2.20E-03	1.45E-01
F6	8.86E-05	5.22E-02	8.86E-05	2.76E-02	1.03E-04	1.35E-01	8.86E-05	8.86E-05	8.86E-05	2.50E-03
F7	8.86E-05	8.86E-05	8.86E-05	8.86E-05	8.86E-05	2.50E-03	8.86E-05	8.86E-05	8.86E-05	4.00E-02
F8	8.86E-05	8.86E-05	8.86E-05	8.86E-05	1.03E-04	8.86E-05	8.86E-05	8.86E-05	2.96E-01	1.71E-03
F9	8.86E-05	1.03E-04	8.86E-05	8.86E-05	8.97E-03	1.40E-04	8.86E-05	8.86E-05	8.86E-05	7.53E-01
F10	8.86E-05	8.86E-05	8.86E-05	8.86E-05	3.77E-04	8.86E-05	8.86E-05	8.86E-05	8.86E-05	8.86E-05
F11	8.86E-05	8.86E-05	8.86E-05	8.86E-05	8.86E-05	8.86E-05	8.86E-05	8.86E-05	3.35E-03	1.00E+00
F12	8.86E-05	8.86E-05	8.86E-05	8.86E-05	8.86E-05	4.00E-02	2.06E-02	8.86E-05	8.86E-05	8.86E-05
F13	8.86E-05	8.86E-05	8.86E-05	8.52E-01	1.40E-04	6.81E-01	1.20E-04	8.86E-05	8.86E-05	8.97E-03
F14	8.86E-05	1.55E-04	2.22E-03	5.75E-01	4.05E-03	7.80E-04	7.80E-04	1.40E-04	3.90E-04	2.28E-02
F15	8.86E-05	8.86E-05	1.17E-01	8.86E-05	8.86E-05	5.17E-04	9.11E-01	8.86E-05	1.20E-04	1.40E-04
F16	8.86E-05	8.86E-05	8.86E-05	8.86E-05	8.86E-05	8.86E-05	8.86E-05	2.50E-03	8.86E-05	7.65E-01
F17	8.86E-05	4.05E-03	8.86E-05	8.86E-05	8.86E-05	8.86E-05	8.86E-05	8.86E-05	8.86E-05	4.38E-02
F18	1.37E-02	3.90E-04	8.86E-05	8.86E-05	3.13E-01	8.86E-05	8.86E-05	1.17E-01	6.27E-01	4.05E-03
F19	7.94E-01	7.94E-01	8.86E-05	6.20E-02	8.86E-05	7.94E-01	8.86E-05	8.86E-05	1.16E-03	5.50E-01
F20	6.00E-01	7.93E-01	1.00E+00	2.51E-02	8.86E-05	6.18E-02	3.12E-01	6.01E-01	1.00E-02	8.86E-05
F21	1.91E-01	6.74E-02	1.17E-01	9.70E-01	2.28E-02	4.55E-03	8.59E-02	2.63E-01	6.74E-02	1.08E-01
F22	1.71E-03	1.00E-02	1.51E-03	6.20E-02	1.45E-01	8.86E-05	1.45E-01	1.00E+00	3.13E-01	2.19E-04
F23	4.05E-03	6.20E-02	7.31E-02	7.80E-04	3.90E-04	1.37E-02	3.13E-01	1.00E-02	1.45E-01	2.54E-04

TABLE 9. Average ranking of all the algorithms based on Friedman test.

	BA	BBO	CSA	DA	GA	GWO	PSO	SBO	EGWO	AGWO	GWOCSA
F1	13	9.3	10.9	9.05	12	4	6.7	8.7	3	2	**1**
F2	13	9.9	11.95	10.43	**1.63**	5	8.1	8.25	4	2.95	1.95
F3	12.95	12.05	8.75	6.5	8.8	4.55	7.55	10.9	4.3	**2.2**	2.55
F4	12.35	12.65	9.3	7	7.95	3.25	6.8	9.25	5.4	1.95	**1.05**
F5	13	8.5	11.45	7.7	9.6	2.95	6.4	9.8	4	3.4	**2.4**
F6	13	5	11.9	7.55	10.2	5.65	**1.75**	3.6	8.5	6.95	5.95
F7	13	10.7	8.7	7.15	6.85	3.1	7.3	11.95	5.1	2.6	**1.75**
F8	**1.75**	2	6.3	11.05	11.95	6.35	4.9	6.5	5.8	9.4	9.5
F9	10.65	5.3	8.75	7.6	4.2	5	9.5	9.45	13	**1.85**	**1.85**
F10	13	6.4	11.5	9.95	7.6	4	8.55	7.95	3.65	2.2	**1.2**
F11	12.95	12.05	10.95	8.6	9.8	4.65	5.9	8.65	4.25	**1.73**	**1.73**
F12	13	10.8	11.05	7	7.9	2.6	3.55	9.45	8.2	3.45	**1.8**
F13	13	3.1	11.85	6.2	5	7.35	3.7	**2.65**	10.3	7.95	7.35
F14	11.33	10.4	**2.6**	3.43	6.38	7.9	8.38	10.18	8.5	6.93	5.08
F15	10.1	10.15	**2.2**	9.1	12.2	5.2	3.1	9.05	7.85	7.75	2.8
F16	6.38	11.9	**3.83**	4.1	13	7.95	3.83	7.73	6.63	10.4	10.45
F17	4.6	9.25	**2.85**	3.03	12.9	8.05	2.85	5.58	6.6	9.8	10.55
F18	4.7	9.95	**3.83**	4	7.5	3.83	3.83	9.73	9.1	9.75	8.8
F19	8.5	8.5	**1.7**	5.5	12.95	8.5	1.7	3.35	5.2	8.45	7.75
F20	5.98	5.83	4.35	6.8	12.95	6.95	**3.25**	3.45	6.5	9.35	5.95
F21	7.45	9.3	**3.43**	5.4	9.1	5.15	7.53	7.7	8.9	7.25	6.5
F22	9.6	9.18	**2.05**	5.78	7.48	4.75	5.78	5.5	7.85	8	7.2
F23	8.55	7.15	**2.95**	7.15	8.95	5.2	5.13	8.25	7.9	7	6.2
Average	231.84	199.36	163.14	160.07	206.89	121.93	126.08	177.62	154.53	133.31	**111.36**

on more than 60% of the multimodal benchmark functions. Nevertheless, some algorithms obtain better results compared to GWOCSA in some special cases, however, GWOCSA has an overall edge in terms of overall performance. Furthermore, it is worthwhile to mention that GWOCSA shows superior performance in comparison to EGWO and AGWO

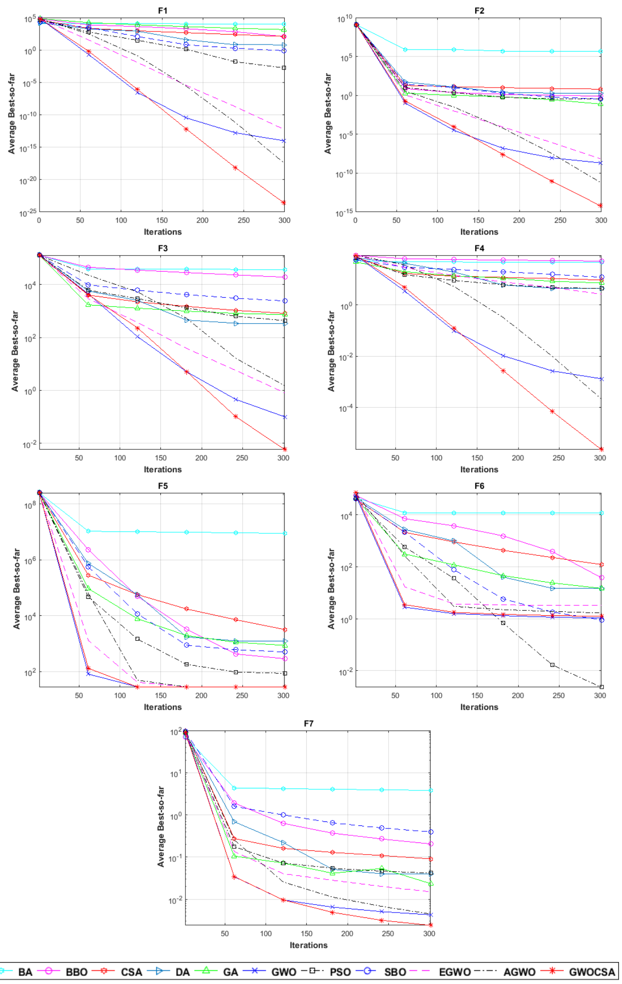

FIGURE 4. Convergence curves of the unimodal functions.

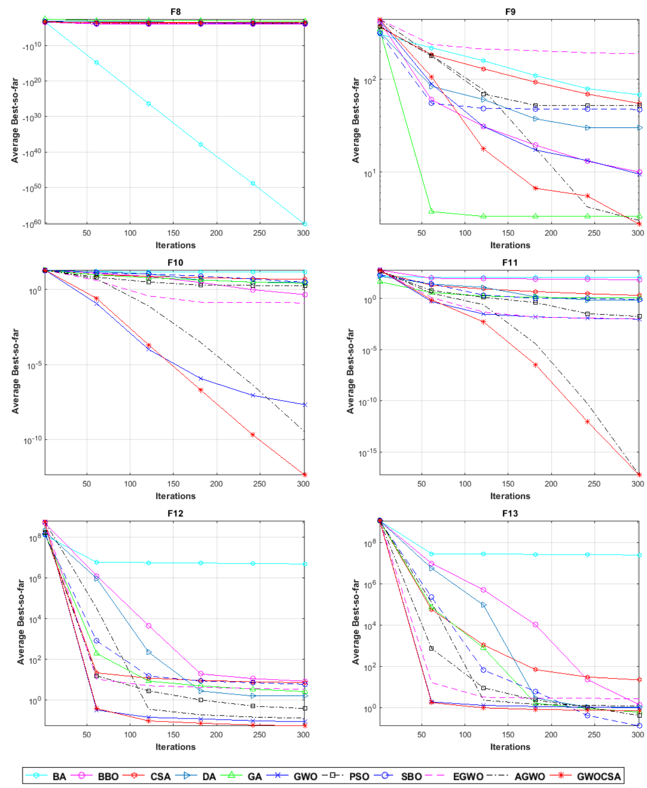

FIGURE 5. Convergence curves of the multimodal functions.

in almost all of the cases. This proves that the hybridization of GWO and CSA has significantly improved the global searching ability and stability. Additionally, Figs. 5 and 6 demonstrate the convergence behavior of all the algorithms on multimodal benchmark functions. As it can be observed from these figures that GWOCSA converges fast compared to other algorithms and achieves optima of most functions which demonstrates the strong exploitation ability in the later stages of optimization as well as the capability to keep the diverse distribution of population during the search process. According to the above observations, it can be concluded that the performance of the proposed GWOCSA is superior to the other ten algorithms when utilized to solve most of these optimization problems.

A nonparametric statistical test, Wilcoxon's rank-sum test [71] is conducted in order is to determine whether the proposed GWOCSA algorithm provides a significant improvement compared to other algorithms or not. The test was carried out using the results of the proposed GWOCSA in each benchmark function and compared with each of the other algorithms at 5% significance. Table 8 presents the *p*-values obtained by the test, where the *p*-values less than 0.05 signify that the null hypothesis is rejected, i.e., there is a significant difference at a level of 5%. On the contrary, the *p*-values (greater than 0.05) are underlined which mean that there is no significant difference between the compared values. It can be analyzed from the results of Table 8 that in most of the comparisons the *p*-values are smaller than 0.05 which verify that the improvement achieved by the proposed GWOCSA is statistically significant on the majority of the benchmark functions.

Table 9 demonstrates the results obtained from the Friedman's test. The aim of the test is to determine whether the proposed GWOCSA algorithm provides a significant improvement compared to the other algorithms. In this test, a rank is assigned on the basis of performance of the algorithm; this means smaller the rank, better the algorithm. As can be seen from Table 9, the GWOCSA algorithm has demonstrated better or similar results compared to the other algorithms for most of the unimodal and multimodal benchmark functions, except for F2, F3, F6, F8 and F13. But for GWOCSA, the rank of these benchmark functions is close to the first rank achieved by other algorithms which indicates GWOCSA gives solutions very close to global optima in these functions. Though some algorithms perform better than the GWOCSA algorithm in some fixed dimensional multimodal benchmark functions, however, GWOCSA algorithm has an overall edge in terms of performance. It can be analyzed from the results of this table that the proposed GWOCSA obtained the best average rank in comparison

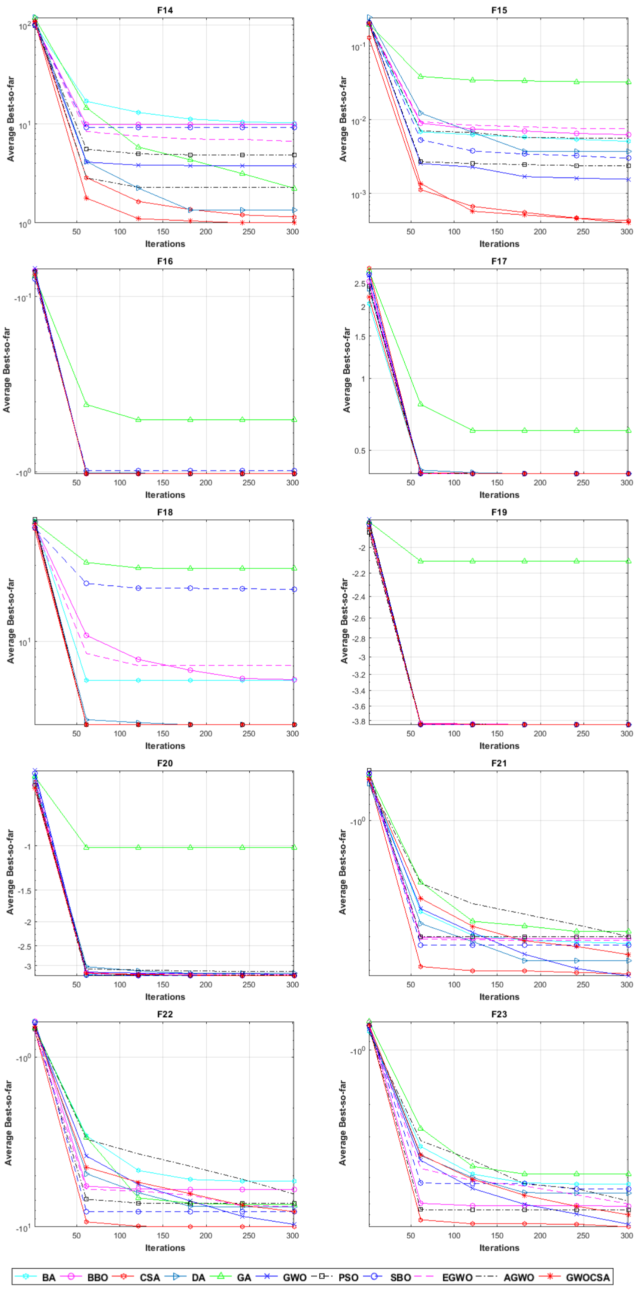

FIGURE 6. Convergence curves of the fixed dimensional functions.

to GWO, CSA and other algorithms. This means that the results of GWOCSA are significantly better than the other algorithms.

B. FEATURE SELECTION EXPERIMENTS

Feature selection problem can be considered as a multi-objective optimization problem in which two opposing goals are to be accomplished; selecting a minimum number of features and achieving maximum classification accuracy. In the feature selection problem, that solution is considered best which contains a minimal number of features along with the highest classification accuracy. Here, several experiments have been performed over twenty-one distinct UCI data sets and the performance of proposed hybrid GWOCSA has been compared with state-of-the-art feature selection approaches. The details of twenty-one data sets have been depicted in Table 10 [22]. These data sets have been selected so that they represent a various number of features and tuples on which the proposed approach needs to be tested [22], [23]. Above all, the selected data sets have a huge search space so that the testing of the optimization algorithm can be performed appropriately. Each dataset is divided in a way as done in cross-validation methods [72].

TABLE 10. Data sets used in the current study.

D. No.	Name	No. of features	No. of samples
D1	Breastcancer	9	699
D2	BreastEW	30	569
D3	Clean1	166	476
D4	Clean2	166	6598
D5	CongressEW	16	435
D6	Exactly	13	1000
D7	Exactly2	13	1000
D8	HeartEW	13	270
D9	IonosphereEW	34	351
D10	KrvskpEW	36	3196
D11	Lymphography	18	148
D12	M-of-n	13	1000
D13	PenglungEW	325	73
D14	Semeion	265	1593
D15	SonarEW	60	208
D16	SpectEW	22	267
D17	Tic-tac-toe	9	958
D18	Vote	16	300
D19	WaveformEW	40	5000
D20	WineEW	13	178
D21	Zoo	16	101

As already discussed, the new positions of the search agents will have continuous solutions and therefore these continuous values must be transformed into corresponding binary values. This conversion is performed by applying squashing of continuous solutions in each dimension using a Sigmoidal (S-shaped) transfer function [73] which will force the search agents to move in a binary search space as shown in Eq. (14).

$$S = \frac{1}{1 + e^{-{x_i}^k}(t)} \tag{14}$$

where ${x_i}^k$ is the continuous-valued position of i^{th} search agent in k^{th} dimension at iteration t.

The output from the S-shaped transfer function is still in a continuous manner and henceforth it has to be the threshold to reach the binary-valued one. The S-shape functions map the infinite input smoothly to a finite output. The commonly stochastic threshold is applied as mentioned in Eq. (15) to reach the binary solution in case of sigmoidal function.

$${x_i}^k(t+1) = \begin{cases} 0 & \text{If rand} < S \\ 1 & \text{If rand} \geq S \end{cases} \tag{15}$$

TABLE 11. Classification accuracy of the proposed GWOCSA vs other feature selection algorithms on different data sets.

	BA	BBO	CSA	DA	GA	GWO	PSO	SBO	EGWO	AGWO	GWOCSA
D1	0.9594	0.9623	0.9606	0.9626	0.9597	0.9603	0.9609	0.9674	0.9617	0.9606	**0.9720**
D2	0.9397	0.9453	0.9460	0.9385	0.9488	0.9375	0.9378	0.9425	0.9474	0.9340	**0.9621**
D3	0.8328	0.8782	0.8798	0.8541	0.8697	0.8580	0.8549	0.8748	0.8487	0.8555	**0.8844**
D4	0.9493	0.9539	0.9514	0.9487	0.9423	0.9463	0.9465	**0.9578**	0.9464	0.9478	0.9564
D5	0.9229	0.9367	0.9330	0.9318	0.9413	0.9327	0.9235	0.9505	0.9431	0.9358	**0.9633**
D6	0.6948	0.7540	0.7664	0.7481	0.7306	0.7249	0.7471	0.7340	0.7536	0.7576	**0.9904**
D7	0.7016	0.6924	0.6908	0.7007	0.6940	0.6929	0.6959	0.7096	0.6988	0.6956	**0.7460**
D8	0.7630	0.7822	0.7822	0.7773	0.7867	0.7768	0.7788	0.7926	0.7615	0.7970	**0.8326**
D9	0.8807	0.8807	0.8943	0.8708	0.8938	0.8682	0.8803	0.8989	0.8636	0.8932	**0.9148**
D10	0.8990	0.9374	0.9079	0.9269	0.9215	0.9143	0.9200	0.9362	0.9272	0.9163	**0.9549**
D11	0.7752	0.8000	0.7919	0.7793	0.8164	0.7629	0.7906	0.8182	0.7663	0.7919	**0.8703**
D12	0.8212	0.8804	0.8560	0.8293	0.7988	0.8272	0.8425	0.8632	0.8704	0.8780	**0.9960**
D13	0.8487	0.8162	0.8054	0.8268	0.6721	0.8341	0.8140	0.8432	0.7568	0.8541	**0.8595**
D14	0.9684	0.9719	0.9676	0.9721	0.9757	0.9671	0.9676	**0.9769**	0.9686	0.9709	0.9767
D15	0.8212	0.8712	0.8538	0.8506	0.8750	0.8622	0.8667	0.8942	0.8615	0.8827	**0.9058**
D16	0.7716	0.7985	0.7925	0.8000	0.8097	0.7846	0.7841	0.7985	0.8045	0.8134	**0.8164**
D17	0.7203	0.7683	0.7649	0.7564	0.7609	0.7537	0.7518	0.7683	0.7712	0.7628	**0.7996**
D18	0.9293	0.9173	0.9213	0.9227	0.9333	0.9196	0.9258	0.9347	0.9027	0.9200	**0.9480**
D19	0.6904	0.7254	0.7139	0.7154	0.6921	0.7096	0.7192	0.7207	0.7165	0.7174	**0.7293**
D20	0.9348	0.9663	0.9618	0.9551	0.9536	0.9476	0.9521	0.9685	0.9663	0.9573	**0.9820**
D21	0.9133	0.9373	0.9333	0.9359	0.9294	0.9525	0.9451	0.9686	0.9686	0.9686	**0.9686**
Average	0.8446	0.8655	0.8607	0.8573	0.8526	0.8539	0.8574	0.8723	0.8574	0.8672	**0.9061**

where ${x_i}^k(t)$ indicates the position of i^{th} search agent at iteration t in k^{th} dimension.

As in K-cross-validation, the testing and validation are performed using $k-1$ folds and k^{th} fold is used for testing. The evaluation for each data set is performed $K \times M$ times. Each data set is divided into three parts: training, validation and testing. The classifier is trained using the training part of the data set and then the performance of the classifier is assessed using the validation part of the data set. Finally, the evaluation of the selected features is performed using the testing data set. In the training process, each searching agent is moved to select a feature subset. The proposed feature selection methods have been compared to the various feature selection methods including BA, BBO, CSA, DA, GA, GWO, PSO, SBO, EGWO and AGWO.

C. FITNESS FUNCTION

In this research work, every solution is characterized as a single dimensional vector in which the length of the vector depends on the number of features/attributes in the data set. Every cell of the vector can contain two values, i.e., 1 or 0, where value 1 depicts that the corresponding feature/attribute is chosen whereas value 0 represents that the feature/attribute is not selected. Every solution is assessed by the proposed fitness function which relies on KNN classifier [74] in order to calculate its classification accuracy on the basis of selected features. Keeping in mind the end goal which is to find the balance between the number of attributes and classification accuracy, the fitness function in Eq. (16) is employed in all the optimization algorithms in order to evaluate the solutions.

$$Fitness = \alpha \gamma_R(D) + \beta \frac{|R|}{|N|} \tag{16}$$

where $\gamma_R(D)$ is the classification accuracy of KNN classifier. Furthermore, $|R|$ represents the cardinality of the selected feature subset and $|N|$ represents the total number of features in the original dataset, α and β are two parameters corresponding to the importance of classification quality and subset length, $\alpha \in [0, 1]$ and $\beta = (1-\alpha)$ adopted from [23] and [46]. Each search agent is evaluated using a fitness function and then, its position is updated. This process is done iteratively until the maximum number of iterations is reached.

Table 11 outlines the results of BA, BBO, CSA, DA, GA, GWO, PSO, SBO, EGWO, AGWO and the proposed GWOCSA algorithm in terms of classification accuracy. The best results are highlighted in bold. As it may be observed, GWOCSA performs superior to other metaheuristics in terms of classification accuracy on 19 data sets, except D4 and D14. The proposed approach obtains the highest classification accuracies on high-dimensional data sets such as D4 and D13. Moreover, GWOCSA demonstrated superior performance on data sets having smaller sample size such as D11, D13, D20 and D21. The reason of the better performance is the enhanced exploration capacity of GWOCSA in comparison to other algorithms. It is worthwhile to mention here that the conventional GWO, classical CSA, EGWO and AGWO do not outperform the proposed GWOCSA over any data set. Overall, the average classification accuracy of GWOCSA is the highest and this superior performance proves the compe-

TABLE 12. Average feature length of the proposed GWOCSA vs other feature selection algorithms on different data sets.

	BA	BBO	CSA	DA	GA	GWO	PSO	SBO	EGWO	AGWO	GWOCSA
D1	6	7.6	6.4	6.27	6.1	6.9	5.7	6.6	6.6	5.2	**5**
D2	15.6	26.8	14.4	20	**12.2**	19	18.33	26.2	20.4	19.2	13.8
D3	90	153	85.2	109.67	98.9	109.6	104.93	148.6	117	93	**85**
D4	**78.8**	152	84.6	100.4	94.1	106	109.4	138.2	84.6	103	86.2
D5	8.6	13	8.8	10.87	7.1	9.8	10.8	9.6	11.2	9.8	**5**
D6	7.4	11.2	8.6	10.53	8.1	12.07	9	11.4	9.2	10.8	**6.4**
D7	6.4	11.2	6.8	8.67	7.1	7.53	9.4	8.8	9.8	7.4	**4.6**
D8	7.6	11.2	8.4	9.6	6.6	8.8	9.07	11.2	8.2	8.6	**5**
D9	18.2	29.8	15.6	18	13.5	17.33	19.2	26	21	22.6	**13**
D10	18	31.6	18.4	28.6	18	31.6	25.6	32.4	26	27.4	**18.6**
D11	8.2	14.8	10	12.53	8.9	11.8	11.73	14.2	11.6	10.4	**8**
D12	**6.4**	10.8	8.4	12.13	7.68	11.27	10.87	11.2	11.2	10.2	**6.4**
D13	160.8	283.6	157.8	175.2	**153**	162.8	183.33	260.6	177.4	158.6	165.8
D14	133.4	245.6	**141.6**	193	149.4	203.6	171.6	248.8	194	187.8	142
D15	**28**	53.2	30.2	40.6	30.3	41.6	37.6	52.4	41.6	47	29.6
D16	10.2	20	11.8	14.67	**7**	13.2	12.07	15.2	12	17.8	8
D17	**5**	9	6.6	7.2	5.8	7.53	6.73	8	7.6	7.8	**5**
D18	6.8	13.4	6.8	8.87	5.8	8.47	9.33	11.6	8.8	9.2	**4.6**
D19	23	36	22.4	36	30.4	36.6	35.8	38	35.2	34.6	**18.4**
D20	8.2	11.4	8	9.53	6.73	10.73	10.07	10.4	9.6	10.6	**6.4**
D21	8.8	12.6	8.2	11.47	5.35	12.4	11.8	12.4	12.8	11.2	**5.2**
Average	31.21	55.13	31.86	40.18	32.48	40.41	39.16	52.47	39.80	38.68	**30.57**

TABLE 13. Statistical mean fitness measure of the proposed GWOCSA vs other feature selection algorithms on different data sets.

	BA	BBO	CSA	DA	GA	GWO	PSO	SBO	EGWO	AGWO	GWOCSA
D1	4.68E-02	4.58E-02	4.11E-02	4.07E-02	4.56E-02	4.70E-02	4.51E-02	3.96E-02	4.52E-02	4.48E-02	**3.33E-02**
D2	6.50E-02	6.31E-02	5.83E-02	6.76E-02	5.48E-02	6.82E-02	6.77E-02	6.57E-02	5.89E-02	7.17E-02	**4.21E-02**
D3	1.71E-01	1.30E-01	1.24E-01	1.51E-01	1.34E-01	1.47E-01	1.50E-01	1.33E-01	1.57E-01	1.49E-01	**1.22E-01**
D4	5.49E-02	5.48E-02	5.32E-02	5.68E-02	6.21E-02	5.96E-02	5.96E-02	5.01E-02	5.82E-02	5.79E-02	**4.84E-02**
D5	8.17E-02	7.08E-02	7.18E-02	7.43E-02	6.26E-02	7.27E-02	8.24E-02	5.51E-02	6.33E-02	6.97E-02	**3.95E-02**
D6	3.08E-01	2.52E-01	2.38E-01	2.57E-01	2.70E-01	2.82E-01	2.57E-01	2.72E-01	2.51E-01	2.48E-01	**1.44E-02**
D7	3.00E-01	3.13E-01	3.11E-01	3.03E-01	3.08E-01	3.10E-01	3.08E-01	2.94E-01	3.06E-01	3.07E-01	**2.55E-01**
D8	2.41E-01	2.24E-01	2.23E-01	2.28E-01	2.16E-01	2.28E-01	2.26E-01	2.14E-01	2.42E-01	2.08E-01	**1.70E-01**
D9	1.23E-01	1.27E-01	1.09E-01	1.33E-01	1.09E-01	1.36E-01	1.24E-01	1.08E-01	1.41E-01	1.12E-01	**8.82E-02**
D10	1.05E-01	7.07E-02	9.63E-02	8.03E-02	8.27E-02	9.37E-02	8.63E-02	7.22E-02	7.93E-02	9.05E-02	**4.98E-02**
D11	2.27E-01	2.06E-01	2.12E-01	2.25E-01	1.87E-01	2.41E-01	2.14E-01	1.88E-01	2.38E-01	2.12E-01	**1.33E-01**
D12	1.82E-01	1.27E-01	1.49E-01	1.78E-01	2.05E-01	1.80E-01	1.64E-01	1.44E-01	1.37E-01	1.29E-01	**9.04E-03**
D13	1.55E-01	1.91E-01	1.98E-01	1.77E-01	**1.29E-01**	1.69E-01	1.90E-01	1.63E-01	2.46E-01	1.49E-01	1.44E-01
D14	3.63E-02	3.71E-02	3.74E-02	3.49E-02	2.90E-02	4.02E-02	3.85E-02	3.22E-02	3.84E-02	3.59E-02	**2.85E-02**
D15	1.82E-01	1.36E-01	1.50E-01	1.55E-01	1.28E-01	1.43E-01	1.38E-01	1.13E-01	1.44E-01	1.24E-01	**9.82E-02**
D16	2.31E-01	2.09E-01	2.11E-01	2.05E-01	1.92E-01	2.19E-01	2.19E-01	2.06E-01	1.99E-01	1.93E-01	**1.85E-01**
D17	2.83E-01	2.39E-01	2.40E-01	2.49E-01	2.43E-01	2.52E-01	2.53E-01	2.38E-01	2.35E-01	2.43E-01	**2.04E-01**
D18	7.42E-02	9.02E-02	8.21E-02	8.21E-02	6.96E-02	8.49E-02	7.93E-02	7.19E-02	1.02E-01	8.50E-02	**5.44E-02**
D19	3.12E-01	2.81E-01	2.89E-01	2.91E-01	3.10E-01	2.97E-01	2.87E-01	2.86E-01	2.89E-01	2.88E-01	**2.73E-01**
D20	7.08E-02	4.21E-02	4.40E-02	5.18E-02	5.12E-02	6.02E-02	5.52E-02	3.92E-02	4.08E-02	5.04E-02	**2.27E-02**
D21	9.14E-02	7.00E-02	7.11E-02	7.06E-02	7.32E-02	5.48E-02	6.17E-02	3.88E-02	3.91E-02	3.81E-02	**3.43E-02**

tency of the proposed approach to efficiently find the optima in the search space.

Table 12 shows the average feature length using GWOCSA in comparison to the other algorithms. GWOCSA demonstrates much better performance by selecting less number of features as compared to other approaches. According to the results reported in this table, GWOCSA performed better on majority of the data sets, except D2, D4, D12, D14, D15 and D17. It is worthwhile to mention that in comparison to the original number of features in the data set, there is

TABLE 14. Statistical standard deviation of the proposed GWOCSA vs other feature selection algorithms on different data sets.

	BA	BBO	CSA	DA	GA	GWO	PSO	SBO	EGWO	AGWO	GWOCSA
D1	8.37E-03	6.21E-03	4.61E-03	1.10E-02	5.00E-03	1.10E-02	8.00E-03	4.70E-03	6.19E-03	7.59E-03	**1.46E-03**
D2	9.46E-03	6.30E-03	5.64E-03	1.00E-02	3.00E-03	6.00E-03	6.00E-03	5.70E-03	7.07E-03	1.38E-02	**1.72E-03**
D3	2.18E-02	5.32E-03	1.33E-02	2.10E-02	**8.00E-03**	1.90E-02	2.60E-02	1.05E-02	1.95E-02	1.07E-02	1.25E-02
D4	4.11E-03	3.25E-03	2.57E-03	3.80E-02	1.36E-01	5.10E-02	6.70E-02	4.01E-03	3.32E-03	5.20E-03	**1.34E-03**
D5	4.15E-03	**3.28E-03**	5.93E-03	2.20E-02	1.10E-02	1.30E-02	1.90E-02	7.01E-03	1.14E-02	1.09E-02	5.56E-03
D6	1.22E-01	3.42E-02	7.24E-02	2.80E-02	2.00E-02	2.30E-02	2.90E-02	2.93E-02	9.26E-02	9.66E-02	**1.52E-02**
D7	1.72E-02	9.94E-03	1.48E-02	1.50E-02	1.20E-02	2.20E-02	2.30E-02	5.81E-03	2.79E-02	1.45E-02	**3.08E-03**
D8	1.91E-02	1.08E-02	2.90E-02	1.40E-02	3.80E-02	**2.00E-03**	4.50E-02	9.31E-03	2.89E-02	1.92E-02	2.11E-02
D9	3.89E-02	1.59E-02	2.21E-02	4.10E-02	1.60E-02	3.00E-02	2.80E-02	3.54E-02	2.25E-02	**8.90E-03**	2.49E-02
D10	4.74E-02	1.93E-02	1.57E-02	1.80E-02	5.40E-02	3.00E-02	5.80E-02	9.21E-03	1.41E-02	9.11E-03	**3.54E-03**
D11	4.82E-02	2.21E-02	**6.82E-03**	2.30E-02	1.30E-02	4.00E-02	2.90E-02	3.64E-02	3.60E-02	4.12E-02	2.24E-02
D12	8.77E-02	5.45E-02	2.46E-02	3.00E-02	1.60E-02	3.10E-02	2.70E-02	1.54E-02	3.55E-02	5.49E-02	**5.88E-03**
D13	4.85E-02	7.17E-02	3.46E-02	2.30E-02	**6.00E-03**	2.60E-02	3.40E-02	1.11E-01	4.16E-02	6.43E-02	3.49E-02
D14	4.15E-03	1.09E-02	7.02E-03	1.90E-02	8.00E-03	2.90E-02	1.90E-02	1.86E-03	6.76E-03	**1.64E-03**	4.60E-03
D15	3.62E-02	3.93E-02	1.81E-02	9.00E-03	**3.00E-03**	1.20E-02	1.20E-02	2.24E-02	3.18E-02	1.28E-02	1.70E-02
D16	3.56E-02	4.75E-02	1.25E-02	1.30E-02	**1.10E-02**	3.30E-02	1.20E-02	4.58E-02	1.37E-02	3.12E-02	3.08E-02
D17	5.79E-02	2.91E-02	1.99E-02	3.70E-02	1.50E-02	4.70E-02	5.00E-02	2.64E-02	1.67E-02	1.99E-02	**1.15E-02**
D18	1.93E-02	1.30E-02	1.28E-02	1.80E-02	9.00E-03	2.20E-02	2.80E-02	2.69E-02	2.34E-02	1.44E-02	**3.17E-03**
D19	1.44E-02	7.93E-03	9.23E-03	3.00E-03	4.00E-03	2.00E-03	**1.00E-03**	7.73E-03	7.92E-03	5.94E-03	5.33E-03
D20	2.68E-02	8.20E-03	**6.26E-03**	7.70E-02	1.80E-02	4.60E-02	7.70E-02	1.42E-02	2.19E-02	1.69E-02	1.23E-02
D21	3.48E-02	2.60E-02	7.05E-02	3.00E-03	**1.00E-03**	5.00E-03	2.00E-03	1.66E-02	2.90E-02	2.70E-02	5.10E-02

a significant reduction in the number of features selected by the proposed approach. For instance, the actual number of features in D3 and D4 data sets is 166 whereas the number of features selected by the proposed GWOCSA is 85 and 86.2 respectively. This indicates that the proposed GWOCSA is able to reduce the number of features as well as has the ability to locate the most relevant optimal feature subset. The strength of the proposed hybrid algorithm lies in the enhanced exploration and exploitation capability which allows it to eliminate redundant attributes and then search the high-performance regions of the feature space intensively.

In Tables 13 and 14, the statistical measures (mean, and standard deviation) obtained on different runs of the algorithms on all the data sets are presented. It can be observed from Table 13 that GWOCSA performed better in mean fitness measure on almost all of the data sets whereas in standard deviation fitness measure GWOCSA has shown competitive performance in comparison to other algorithms as shown in Table 14.

Table 15 reports the p-values of GWOCSA in comparison to other metaheuristic algorithms obtained using Wilcoxon's rank sum test. The test is conducted in order to determine whether the difference between the results of proposed GWOCSA and other algorithms' results is significant or not. Specifically, a p-value is returned and if the value of p-value is smaller than 0.05 then it indicates that the results achieved by GWOCSA are significantly different than those of the compared algorithms whereas p-value greater than 0.05 indicates that there is no significant difference exists between results of GWOCSA and the compared algorithms. The worst results of p value where p greater than 0.05 are underlined. It can be easily observed that in most of the comparisons, the p-values obtained using the rank sum test are smaller than 0.05 which prove that the superiority of GWOCSA is statistically significant. These results are consistent with the acquired results from Tables 13 and 14.

The superior results do not mean that the proposed approach GWOCSA can tackle all the optimization problems efficiently. As per the NFL theorem, all optimization algorithms demonstrate identical performance when employed to solve all classes of optimization problems [52]. Undoubtedly, there are some limitations of the proposed algorithm; the first limitation of the GWOCSA is that the calculation of the adaptive balance probability p and control parameter $\vec{a}$ adds an overhead to the computational time of GWOCSA. Second limitation is that a search agent updates its position to a random place in solution space which decreases the performance of the proposed algorithm. Thirdly, the transfer function which is utilized in this study to convert the continuous value to discrete value is of non-time-varying nature which means it cannot adapt the exploration behavior at the beginning of the optimization process when it is required to deal with challenging feature spaces. This will refrain the algorithm to further explore some promising regions inside the feature space. Since the proposed GWOCSA demonstrated competitive performance on function optimization problems as well as feature selection problems, therefore, we suggest GWOCSA to researchers in different fields. The proposed hybrid GWOCSA has a high potential to demonstrate very promising and/or superior results.

TABLE 15. *p* -values obtained from the rank-sum test on different data sets ($p > 0.05$ are underlined).

	BA	BBO	CSA	DA	GA	GWO	PSO	SBO	EGWO	AGWO
D1	4.31E-02	4.31E-02	4.31E-02	2.25E-01	4.31E-02	4.22E-02	4.31E-02	7.96E-02	4.22E-02	4.31E-02
D2	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.22E-02	4.31E-02	4.31E-02
D3	4.31E-02	2.25E-01	6.86E-01	4.31E-02	7.96E-02	4.31E-02	4.31E-02	2.25E-01	4.31E-02	4.31E-02
D4	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	5.00E-01	4.31E-02	4.31E-02
D5	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	7.96E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02
D6	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02
D7	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02
D8	4.31E-02	4.22E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.22E-02	4.31E-02	7.96E-02
D9	4.31E-02	4.31E-02	2.25E-01	4.31E-02	7.82E-02	4.31E-02	2.25E-01	4.31E-02	7.96E-02	1.38E-01
D10	4.31E-02	1.38E-01	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.22E-02	4.31E-02	4.31E-02
D11	4.22E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.22E-02
D12	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02
D13	6.86E-01	4.31E-02	4.31E-02	4.31E-02	2.25E-01	7.96E-02	4.31E-02	4.31E-02	4.31E-02	8.93E-01
D14	4.31E-02	2.25E-01	4.31E-02	4.31E-02	1.38E-01	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02
D15	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	7.96E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02
D16	4.31E-02	4.31E-02	4.31E-02	4.31E-02	2.25E-01	7.96E-02	4.31E-02	4.31E-02	3.45E-01	6.86E-01
D17	4.31E-02	1.38E-01	4.31E-02	7.96E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02
D18	7.96E-02	4.31E-02	4.31E-02	4.31E-02	4.22E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.22E-02
D19	4.31E-02	4.31E-02	4.31E-02	7.96E-02	4.31E-02	4.31E-02	7.96E-02	4.31E-02	4.31E-02	4.31E-02
D20	4.31E-02	4.31E-02	4.31E-02	4.22E-02	4.22E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02	4.31E-02
D21	4.31E-02	4.31E-02	4.31E-02	4.31E-02	1.36E-01	6.86E-01	4.31E-02	5.00E-01	5.00E-01	6.86E-01

VI. CONCLUSION AND FUTURE DIRECTIONS

In this paper, a hybrid algorithm based on Grey Wolf Optimizer with Crow Search Algorithm is proposed to solve function optimization as well as feature selection problem. In order to overcome the limitations of GWO, it is hybridized with CSA which allows the proposed hybrid algorithm to effectively explore the search space. In order to fully utilize the strengths of both the algorithms, an adaptive balance probability is proposed which allows the proposed GWOCSA algorithm to achieve acceleration throughout early steps of optimization process whereas in the later stages of optimization promising solutions will possess a high probability to be exploited. Moreover, in the proposed algorithm the values of control parameter $\vec{a}$ are selected using a nonlinearly decreasing approach rather than a linearly decreasing approach which enhanced the search capacity of the proposed hybrid algorithm. In order to test the efficiency, the results of the hybrid algorithm on 23 benchmark functions and 21 data sets as a feature selection algorithm are compared with ten metaheuristic algorithms, i.e., ALO, CSA, FPA, GA, GWO, PSO, SHO, AGWO and IGWO. The test results based on the benchmark functions demonstrate that the proposed algorithm has a better function optimization ability and a faster convergence speed, and can obtain more satisfactory optimization results in less iterative times. As a feature selection algorithm, the results of the proposed GWOCSA algorithm demonstrate superiority in terms of classification accuracy and the number of optimal features selected compared to other feature selection algorithms. Additionally, the results of Wilcoxon's test and Friedman's test also indicate that the results of the proposed hybrid algorithm are statistically significant compared to the other metaheuristics. In future, the proposed hybrid algorithm can be applied to more practical problems in real-life world scenarios. Furthermore, utilizing the hybrid algorithm as a filter feature selection approach seeking to evaluate the generality of the selected features will be a valuable contribution.

APPENDIX

For feature selection problem, the following performance metrics are utilized to compare the performance of each feature selection approach.

1) CLASSIFICATION ACCURACY

It is one of the major classification metrics that represents the number of instances which are correctly classified by using a particular set of features. The mathematical formulation of this metric can be defined as:

$$Avg\_performance = \frac{1}{M}\sum_{j=1}^{M}\frac{1}{N}\sum_{i=1}^{N} match(C_i, L_i), \quad (17)$$

where M is the number of times the optimization algorithm has been run, N denotes the number of test set points, C_i is output class label for particular data point i, L_i is reference class label for i and *match* is the comparison function that gives output 0 when input labels are different and 1 if the input label matches with each other.

2) STATISTICAL MEAN

It is the average of the fitness values acquired after running all the iterations of the optimization algorithms as given in the following equation:

$$Mean = \frac{1}{M}\sum_{i=1}^{M} g^{*i} \tag{18}$$

where g^{*i} represents the best solution in the i^{th} run.

3) STATISTICAL STANDARD DEVIATION

The variability of different solutions from the mean value can be computed using one of the measures of dispersion called standard deviation. Standard deviation will be low if the solution forms a highly dense clustered and vice-versa and can be formulated as follows:

$$Std\_dev = \sqrt{\frac{1}{M-1}\sum (g^{*i} - Mean)^2} \tag{19}$$

where g^{*i} represents the best solution in the i^{th} run.

4) AVERAGE FEATURE LENGTH

It characterizes the average length of the selected features to the total number of features. The mathematical formulation for the same is given below:

$$Avg\_feature\_length = \frac{1}{M}\sum_{i=1}^{M} size(g^{*i}) \tag{20}$$

where $size(x)$ represents the number of features selected in the testing data set.

5) WILCOXON RANK SUM TEST

It is a nonparametric statistical test used to check whether the results of the proposed approaches are statistically different from other algorithms [71]. This statistical test returns a parameter called p-value which is utilized to verify the significance level of two algorithms.

ACKNOWLEDGEMENT

The authors gratefully acknowledge the anonymous reviewers for providing their constructive comments.

REFERENCES

[1] X.-S. Yang, *Introduction to Mathematical Optimization: From Linear Programming to Metaheuristics*. Cambridge, U.K.: Cambridge International Science Publishing, 2008.
[2] J. Brownlee, *Clever Algorithms: Nature-inspired Programming Recipes*, 1st ed. Raleigh, NC, USA: LuLu, 2011.
[3] X.-S. Yang, A. H. Gandomi, S. Talatahari, and A. H. Alavi, *Metaheuristics in Water, Geotechnical and Transport Engineering*. Waltham, MA, USA: Newnes, 2012.
[4] A. R. Yıldız, "An effective hybrid immune-hill climbing optimization approach for solving design and manufacturing optimization problems in industry," *J. Mater. Process. Technol.*, vol. 209, no. 6, pp. 2773–2780, 2009.
[5] A. H. Gandomi and A. H. Alavi, "Multi-stage genetic programming: A new strategy to nonlinear system modeling," *Inf. Sci.*, vol. 181, no. 23, pp. 5227–5239, 2011.
[6] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micro Mach. Hum. Sci. (MHS)*, Oct. 1995, pp. 39–43.
[7] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Mach. Learn.*, vol. 3, nos. 2–3, pp. 95–99, 1988.
[8] R. Storn and K. Price, "Differential evolution–A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
[9] X. S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization (NICSO)*. Springer, 2010, pp. 65–74.
[10] X-S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *Int. J. Bio-Inspired Comput.*, vol. 2, no. 2, pp. 78–84, 2010.
[11] S. Arora, S. Singh, S. Singh, and B. Sharma, "Mutated firefly algorithm," in *Proc. Int. Conf. Parallel, Distrib. Grid Comput. (PDGC)*, Dec. 2014, pp. 33–38.
[12] S. Arora and S. Singh, "Butterfly optimization algorithm: A novel approach for global optimization," *Soft Comput.*, vol. 23, no. 3, pp. 715–734, 2018.
[13] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
[14] A. Askarzadeh, "A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm," *Comput. Struct.*, vol. 169, pp. 1–12, Jun. 2016.
[15] M. Bennasar, Y. Hicks, and R. Setchi, "Feature selection using joint mutual information maximisation," *Expert Syst. Appl.*, vol. 42, pp. 8520–8532, Sep. 2015.
[16] S. Gu, R. Cheng, and Y. Jin, "Feature selection for high-dimensional classification using a competitive swarm optimizer," *Soft Comput.*, vol. 22, no. 3, pp. 811–822, 2018.
[17] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms," *Appl. Soft Comput.*, vol. 18, pp. 261–276, May 2014.
[18] M. M. Kabir, M. Shahjahan, and K. Murase, "A new local search based hybrid genetic algorithm for feature selection," *Neurocomputing*, vol. 74, no. 17, pp. 2914–2928, 2011.
[19] H. Chen, W. Jiang, C. Li, and R. Li, "A heuristic feature selection approach for text categorization by using chaos optimization and genetic algorithm," *Math. Problems Eng.*, vol. 2013, Nov. 2013, Art. no. 524017.
[20] M. Mafarja and S. Abdullah, "Record-to-record travel algorithm for attribute reduction in rough set theory," *J. Theor. Appl. Inf. Technol.*, vol. 49, no. 2, pp. 507–513, 2013.
[21] M. Mafarja and S. Abdullah, "A fuzzy record-to-record travel algorithm for solving rough set attribute reduction," *Int. J. Syst. Sci.*, vol. 46, no. 3, pp. 503–512, 2015.
[22] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371–381, Jan. 2016.
[23] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary ant lion approaches for feature selection," *Neurocomputing*, vol. 213, pp. 54–65, Nov. 2016.
[24] S. A.-F. Sayed, E. Nabil, and A. Badr, "A binary clonal flower pollination algorithm for feature selection," *Pattern Recognit. Lett.*, vol. 77, pp. 21–27, Jul. 2016.
[25] M. Mafarja and S. Mirjalili, "Whale optimization approaches for wrapper feature selection," *Appl. Soft Comput.*, vol. 62, pp. 441–453, Jan. 2018.
[26] S. Arora and P. Anand, "Binary butterfly optimization approaches for feature selection," *Expert Syst. Appl.*, vol. 116, pp. 147–160, Feb. 2019.
[27] E.-G. Talbi, *Metaheuristics: From Design to Implementation*, vol. 74. Hoboken, NJ, USA: Wiley, 2009.
[28] E.-G. Talbi, "A taxonomy of hybrid metaheuristics," *J. Heuristics*, vol. 8, no. 5, pp. 541–564, 2002.
[29] Y.-T. Kao and E. Zahara, "A hybrid genetic algorithm and particle swarm optimization for multimodal functions," *Appl. Soft Comput.*, vol. 8, no. 2, pp. 849–857, 2008.
[30] J.-T. Tsai, T.-K. Liu, and J.-H. Chou, "Hybrid Taguchi-genetic algorithm for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 4, pp. 365–377, Aug. 2004.
[31] D. Jitkongchuen, "A hybrid differential evolution with grey wolf optimizer for continuous global optimization," in *Proc. 7th Int. Conf. Inf. Technol. Electr. Eng. (ICITEE)*, Oct. 2015, pp. 51–54.
[32] E. Nabil, "A modified flower pollination algorithm for global optimization," *Expert Syst. Appl.*, vol. 57, pp. 192–203, Sep. 2016.
[33] M. A. Tawhid and A. F. Ali, "A hybrid grey wolf optimizer and genetic algorithm for minimizing potential energy function," *Memetic Comput.*, vol. 9, no. 4, pp. 347–359, 2017.
[34] T. Jayabarathi, T. Raghunathan, B. Adarsh, and P. N. Suganthan, "Economic dispatch using hybrid grey wolf optimizer," *Energy*, vol. 111, pp. 630–641, Sep. 2016.

[35] N. Singh and S. Singh, ''Hybrid algorithm of particle swarm optimization and grey wolf optimizer for improving convergence performance,'' *J. Appl. Math.*, vol. 2017, Nov. 2017, Art. no. 2030489.
[36] P. J. Gaidhane and M. J. Nigam, ''A hybrid grey wolf optimizer and artificial bee colony algorithm for enhancing the performance of complex systems,'' *J. Comput. Sci.*, vol. 27, pp. 284–302, Jul. 2018.
[37] X. Zhang, Q. Kang, J. Cheng, and X. Wang, ''A novel hybrid algorithm based on biogeography-based optimization and grey wolf optimizer,'' *Appl. Soft Comput.*, vol. 67, pp. 197–214, Jun. 2018.
[38] A. E. Hassanien, R. M. Rizk-Allah, and M. Elhoseny, ''A hybrid crow search algorithm based on rough searching scheme for solving engineering optimization problems,'' *J. Ambient Intell. Hum. Comput.*, pp. 1–25, Jun. 2018. doi: 10.1007/s12652-018-0924-y.
[39] I.-S. Oh, J.-S. Lee, and B.-R. Moon, ''Hybrid genetic algorithms for feature selection,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1424–1437, Nov. 2004.
[40] E.-G. Talbi, L. Jourdan, J. Garcia-Nieto, and E. Alba, ''Comparison of population based metaheuristics for feature selection: Application to microarray data classification,'' in *Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl. (AICCSA)*, Mar. 2008, pp. 45–52.
[41] P. Moradi and M. Gholampour, ''A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy,'' *Appl. Soft Comput.*, vol. 43, pp. 117–130, Jun. 2016.
[42] M. E. Basiri and S. Nemati, ''A novel hybrid ACO-GA algorithm for text feature selection,'' in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2009, pp. 2561–2568.
[43] R. S. Babatunde, S. O. Olabiyisi, and E. O. Omidiora, ''Feature dimensionality reduction using a dual level metaheuristic algorithm,'' *Optimization*, vol. 7, no. 1, pp. 49–52, 2014.
[44] J. Jona and N. Nagaveni, ''Ant-cuckoo colony optimization for feature selection in digital mammogram,'' *Pakistan J. Biol. Sci.*, vol. 17, no. 2, p. 266, 2014.
[45] E. Zorarpacı and S. A. Özel, ''A hybrid approach of differential evolution and artificial bee colony for feature selection,'' *Expert Syst. Appl.*, vol. 62, pp. 91–103, Nov. 2016.
[46] M. M. Mafarja and S. Mirjalili, ''Hybrid whale optimization algorithm with simulated annealing for feature selection,'' *Neurocomputing*, vol. 260, pp. 302–312, Oct. 2017.
[47] S. Saremi, S. Mirjalili, and A. Lewis, ''Grasshopper optimisation algorithm: Theory and application,'' *Adv. Eng. Softw.*, vol. 105, pp. 30–47, Mar. 2017.
[48] A. Lewis, S. Mostaghim, and M. Randall, ''Evolutionary population dynamics and multi-objective optimisation problems,'' in *Multi-Objective Optimization in Computational Intelligence: Theory and Practice*. Hershey, PA, USA: IGI Global, 2008, pp. 185–206.
[49] M. Mafarja *et al.*, ''Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems,'' *Knowl.-Based Syst.*, vol. 145, pp. 25–45, Apr. 2018.
[50] I. Boussaïd, J. Lepagnot, and P. Siarry, ''A survey on optimization metaheuristics,'' *Inf. Sci.*, vol. 237, pp. 82–117, Jul. 2013.
[51] G. Chandrashekar and F. Sahin, ''A survey on feature selection methods,'' *Comput. Elect. Eng.*, vol. 40, no. 1, pp. 16–28, Jan. 2014.
[52] D. H. Wolper and W. G. Macready, ''No free lunch theorems for optimization,'' *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.
[53] N. Muangkote, K. Sunat, and S. Chiewchanwattana, ''An improved grey wolf optimizer for training q-Gaussian radial basis functional-link nets,'' in *Proc. Int. Comput. Sci. Eng. Conf. (ICSEC)*, Jul. 2014, pp. 209–214.
[54] D. Zou, H. Liu, L. Gao, and S. Li, ''A novel modified differential evolution algorithm for constrained optimization problems,'' *Comput. Math. Appl.*, vol. 61, no. 6, pp. 1608–1623, 2011.
[55] G. I. Sayed, A. E. Hassanien, and A. T. Azar, ''Feature selection via a novel chaotic crow search algorithm,'' *Neural Comput. Appl.*, vol. 31, no. 1, pp. 171–188, 2017.
[56] D. Gupta, J. J. P. C. Rodrigues, S. Sundaram, A. Khanna, V. Korotaev, and V. H. C. de Albuquerque, ''Usability feature extraction using modified crow search algorithm: A novel approach,'' *Neural Comput. Appl.*, pp. 1–11, Aug. 2018. doi: 10.1007/s00521-018-3688-6.
[57] D. Oliva, S. Hinojosa, E. Cuevas, G. Pajares, O. Avalos, and J. Gálvez, ''Cross entropy based thresholding for magnetic resonance brain images using crow search algorithm,'' *Expert Syst. Appl.*, vol. 79, pp. 164–180, Aug. 2017.
[58] L. dos Santos Coelho, C. Richter, V. C. Mariani, and A. Askarzadeh, ''Modified crow search approach applied to electromagnetic optimization,'' in *Proc. IEEE Conf. Electromagn. Field Comput. (CEFC)*, Nov. 2016, p. 1.
[59] N. Mittal, U. Singh, and B. S. Sohi, ''Modified grey wolf optimizer for global engineering optimization,'' *Appl. Comput. Intell. Soft Comput.*, vol. 2016, p. 8, Mar. 2016.
[60] W. Long, J. Jiao, X. Liang, and M. Tang, ''An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization,'' *Eng. Appl. Artif. Intell.*, vol. 68, pp. 63–80, Feb. 2018.
[61] D. Simon, ''Biogeography-based optimization,'' *IEEE Trans. Evol. Comput.*, vol. 12, no. 6, pp. 702–713, Dec. 2008.
[62] S. Mirjalili, ''Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems,'' *Neural Comput. Appl.*, vol. 27, no. 4, pp. 1053–1073, 2016.
[63] J. Holland and D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA, USA: Addison-Wesley, 1989.
[64] R. Eberhart and J. Kennedy, ''Particle swarm optimization,'' in *Proc. IEEE Int. Conf. Neural Netw.*, Nov. 1995, pp. 1942–1948.
[65] S. H. S. Moosavi and V. K. Bardsiri, ''Satin bowerbird optimizer: A new optimization algorithm to optimize anfis for software development effort estimation,'' *Eng. Appl. Artif. Intell.*, vol. 60, pp. 1–15, Apr. 2017.
[66] H. Joshi and S. Arora, ''Enhanced grey wolf optimization algorithm for global optimization,'' *Fundam. Inform.*, vol. 153, no. 3, pp. 235–264, 2017.
[67] M. H. Qais, H. M. Hasanien, and S. Alghuwainem, ''Augmented grey wolf optimizer for grid-connected PMSG-based wind energy conversion systems,'' *Appl. Soft Comput.*, vol. 69, pp. 504–515, Aug. 2018.
[68] A. R. A. Alsewari and K. Z. Zamli, ''Design and implementation of a harmony-search-based variable-strength t-way testing strategy with constraints support,'' *Inf. Softw. Technol.*, vol. 54, no. 6, pp. 553–568, 2012.
[69] S. Mirjalili and A. Lewis, ''The whale optimization algorithm,'' *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.
[70] S. Arora and P. Anand, ''Chaotic grasshopper optimization algorithm for global optimization,'' *Neural Comput. Appl.*, pp. 1–21, Jan. 2018. doi: 10.1007/s00521-018-3343-2.
[71] J. Derrac, S. García, D. Molina, and F. Herrera, ''A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms,'' *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, 2011.
[72] J. Franklin, ''The elements of statistical learning: Data mining, inference and prediction,'' *Math. Intell.*, vol. 27, no. 2, pp. 83–85, 2005.
[73] S. Mirjalili and S. Z. M. Hashim, ''BMOA: Binary magnetic optimization algorithm,'' *Int. J. Mach. Learn. Comput.*, vol. 2, no. 3, p. 204, 2012.
[74] N. S. Altman, ''An introduction to kernel and nearest-neighbor nonparametric regression,'' *Amer. Statist.*, vol. 46, no. 3, pp. 175–185, 1992.

SANKALAP ARORA has over seven years of teaching experience. He is currently an Assistant Professor with the Department of Computer Science and Engineering, DAV University, Jalandhar. His research interest includes optimization algorithms and their applications in the field of machine learning. He has published various research papers in several reputed international conferences and journals. He has been a member of the technical program committee of various international conferences including IEEE/Springer conferences. He has been a reviewer of various journals.

HARPREET SINGH received the B.Tech. degree in computer science and engineering from Guru Nanak Dev University, Amritsar, the M.Tech. degree in computer science and engineering from the National Institute of Technology, Jalandhar, and the Ph.D. degree from Guru Nanak Dev University. He is actively pursuing research in the fields of data mining and optimization algorithms.

MANIK SHARMA received the Ph.D. degree in distributed query optimization. He has qualified to the M.C.A., UGC-NET. He has over 12 years of teaching experience. He is currently involved in the area of data mining, machine learning, and soft computing with a specialization in the diagnosis of diseases and modeling of DSS query optimizer. The thrust area is to improve the predictive accuracy in early diagnosis of several fatal human disorders. He has published over 20 research papers in various international reputed journals and in Scopus and Thomson Reuters. He has also presented over 10 research articles in various national and international UGC and IEEE conferences.

SANJEEV SHARMA is currently an Assistant Professor and the Head of the Department Computer Science and Automation, DAV University, Jalandhar. He is currently guiding four Ph.D. students in the fields of the NLP and machine learning. His specialization is in natural language processing (NLP) which is the sub-branch of artificial intelligence. He has over 16 years of teaching, administrative, and industrial and research experience. He has published over 34 research papers in international journals. He has developed many basic NLP tools like POS Tagger, Clause Identifier, Sentence Identifier, Unknown Word Guesser, and Morph, and one advance tool, i.e., Grammar Checker to detect and rectify/suggest the grammatical errors in compound and complex sentences in the Punjabi language.

PRIYANKA ANAND received the B.Tech. and M.Tech. degrees in computer science and engineering from Lovely Professional University, Phagwara, Punjab, where she is currently an Assistant Professor. She has over seven years of teaching experience. She has published several research papers in international journals, conferences in the fields of optimization algorithms and machine learning.