New Fitness Functions in Binary Harris Hawks Optimization for Gene Selection in Microarray Datasets

Ruba Abu Khurma[1], Pedro A. Castillo[2], Ahmad Sharieh[1] and Ibrahim Aljarah[1]
[1]*King Abdullah II School for Information Technology, The University of Jordan, Amman, Jordan*
[2]*Department of Computer Architecture and Computer Technology, ETSIIT and CITIC, University of Granada, Granada, Spain*

Keywords: Swarm Intelligence, Harris Hawks Optimization, Gene Selection, Microarray Technology.

Abstract: Gene selection (GS) is a challenging problem in medical applications. This is because of the availability of a large number of genes and a limited number of patient's samples in microarray datasets. Selecting the most relevant genes is a necessary pre-processing step for building reliable cancer classification systems. This paper proposes two new fitness functions in Binary Harris Hawks Optimization (BHHO) for GS. The main objective is to select a small number of genes and achieve high classification accuracy. The first fitness function balances between the classification performance and the number of genes. This is done by using a weight that increases linearly throughout the optimization process. The second fitness function is applied across two-stages. The first stage optimizes the classification performance only while the second stage takes into consideration the number of genes. K-nearest neighbor (K-nn) is used to evaluate the proposed approaches on ten microarray data sets. The results show that the proposed fitness functions can achieve better classification results compared with the fitness function that takes into account only the classification performance. Besides, they outperform three other wrapper-based methods in most of the cases. The second fitness function outperforms the first fitness function across most of the datasets based on classification accuracy and the number of genes.

1 INTRODUCTION

Microarray technology is a major application in medicine and bioinformatics. In recent years, gene expression datasets have been used for the diagnosis of many diseases such as cancer disease. The gene expression data sets cause a challenging problem for data mining tasks (e.g classification). This is because they are coded by a large number of genes and a limited number of instances that represent the clinical patient status (Alomari et al., 2018). The large dimensionality problem, also known as the curse of dimensionality problem has many negative consequences on the classification system. The existence of irrelevant and redundant genes reduces the effectiveness of the generalization process, complicates the learned model, increases the learning time, makes the identification of a disease a difficult task and increases the cost of the biological classification system due to increasing the demand for specialized resources.

Gene selection (GS) is a data mining method that is used to simplify the large scale gene expression data sets by retaining the most informative genes. These are considered the key marker in the identification of a disease (Chuang et al., 2011). The noisy genes such as irrelevant and redundant genes are discarded and eliminated from the training process. This can simplify the learning model, speed up the learning process, and potentially increase the performance of disease identification (Mohamad et al., 2011). The output of the GS process is a compact gene expression data that can be used in the testing stage to produce the final decision about the clinical status of a patient.

GS compromises the search and evaluation processes. According to evaluation, the GS methods are commonly classified into filter and wrapper methods (Khurma et al., 2020). The filter methods (e.g Information gain) use the intrinsic characteristics of the dataset. Wrapper methods uses a learning algorithm in GS process to perform an internal learning process.

The search algorithm examines the gene space to find the optimal subset of genes that contains the most useful genes in the diagnosis of a disease. Traditionally, the complete search methods generate the entire gene space and exhaustively traverse all the gene subsets. This leads to an exponential running time, which makes GS an Np-hard problem.

Meta-heuristic algorithms (MH) are stochastic

Khurma, R., Castillo, P., Sharieh, A. and Aljarah, I.
New Fitness Functions in Binary Harris Hawks Optimization for Gene Selection in Microarray Datasets.
DOI: 10.5220/0010021601390146
In *Proceedings of the 12th International Joint Conference on Computational Intelligence (IJCCI 2020)*, pages 139-146
ISBN: 978-989-758-475-6

search methods that have been widely used to mitigate the GS process (Khurma. et al., 2020). They initialize random solutions at the beginning of the search and iteratively evaluate and update them until a stopping criterion is satisfied. MH algorithms include the Swarm Intelligence (SI) algorithm (Khurma et al., 2020).

SI algorithms are inspired by the natural behavior of creatures such as a flock of birds, swarm of wolves, school of fish. Swarm-based systems share information between the individuals of a swarm to survive. Common examples of SI algorithms include Particle Swarm Optimization (PSO) (Emary et al., 2016), Grey Wolves Optimization (GWO) (Zawbaa et al., 2018), Bat Algorithm (BA) (Al-Betar et al., 2020), and Cuckoo Search (CS) (Moghadasian and Hosseini, 2014).

The search process in SI algorithms has mainly two phases: exploration and exploitation. In exploration, the individuals search globally in the gene search space to find the most promising region that may include the optimal solution. In exploitation, the individuals search locally in the found region to approach to the optimal solution.

Many SI algorithms have been proposed and modified in the literature to increase the performance of GS process for classification of diseases. In (Tran et al., 2014), the Binary Particle Swarm optimization algorithm (BPSO) was modified by adopting the reset strategy to allow the stagnated best solution to jump from local minima. The BPSO was modified by Boolean algebra operation in (Emary et al., 2016). The BPSO was modified using the binary quantum operator in (Xi et al., 2016). The new model called BQPSO applied a sampling around the personal best, then used the average of the sampled points to update the current solution. A hybrid model called GWO-ALO integrated the GWO and Ant Lion Optimization (ALO) in (Zawbaa et al., 2018). The main objective was to exploit the global search ability of the GWO and the local search performance of the ALO. A model compromised of the CS algorithm, Mutual Information (MI), entropy filters, and artificial neural network (ANN) was proposed in (Moghadasian and Hosseini, 2014). In (Al-Betar et al., 2020), a hybrid filter/wrapper, called rMRMR-MBA was proposed based on robust Minimum Redundancy Maximum Relevancy (rMRMR) filter and a modified bat algorithm (MBA) using TRIZ optimisation operators.

Harris Hawks Optimization (HHO) is a new SI algorithm proposed in 2019 (Heidari et al., 2019). The HHO simulates the hunting behavior of Harris' hawks in nature. The extensive experimental comparisons in (Heidari et al., 2019) showed that HHO outperformed well-regarded optimization algorithms when they were investigated on unconstrained, unimodal, multi-modal, and composition problems. A wide range of applications adopted the HHO algorithm (Thaher et al., 2020). However, HHO has never been used in the medical and bioinformatics applications. This paper investigates for the first time the performance of the HHO algorithm in solving the GS problem. The accuracy and the size of the selected gene subsets are used as evaluation measurements. The stability of the clinical classification system is determined by noticing the standard deviation of the average accuracy results across 30 runs of the algorithm.

This paper aims to develop a new fitness function in HHO for GS in microarray data sets. The overall goal is to enhance the classification performance of cancer diseases. The expectation is to determine the smallest subset of relevant genes that can increase the classification performance. To achieve this goal, this paper proposes two new fitness functions in HHO for solving the GS problem. The new fitness functions will be evaluated on ten benchmark microarray data sets with different numbers of genes and samples. Specifically, we will investigate:

- The performance of HHO with a fitness function that considers the classification performance only.
- The performance of HHO with a fitness function that considers the classification performance and number of genes simultaneously.
- The performance of HHO with a two-stage fitness function that considers the classification performance first then takes into account the number of selected genes.

The remainder of the paper is organized as follows: Harris Hawks Optimization is presented in Section 2. Section 3 describes the HHO for GS. Section 4 describes the two proposed BHHO based GS approaches with new fitness functions. Section 5 describes the experimental design. Section 6 presents the experimental results with discussions. Comparisons with wrapper-based GS methods are provided in Section 7. Finally, Section 8 provides conclusions and future work.

2 HARRIS HAWKS OPTIMIZATION

HHO has two phases of exploration and four phases of exploitation.

2.1 Exploration Phase

In HHO, the Harris' hawks perch randomly on some locations and wait to detect a prey based on two strategies. By considering equal chance q for each perching strategy, they perch based on the positions of other family members and the rabbit, which is modeled in Eq. (1) for the condition of $q < 0.5$, or perch on random tall trees , which is modeled in Eq. (1) for condition of $q \geq 0.5$.

$$X(t+1) = \begin{cases} X_{rand}(t) - r_1 \left| X_{rand}(t) - 2r_2 X(t) \right| & q \geq 0.5 \\ (X_{rabbit}(t) - X_m(t)) - r_3(LB + r_4(UB - LB)) & o.w \end{cases} \quad (1)$$

where $X(t+1)$ is the position vector of hawks in the next iteration t, $X_{rabbit}(t)$ is the position of rabbit, $X(t)$ is the current position vector of hawks, r_1, r_2, r_3, r_4, and q are random numbers inside (0,1), which are updated in each iteration, LB and UB show the upper and lower bounds of variables, $X_{rand}(t)$ is a randomly selected hawk from the current population, and X_m is the average position of the current population of hawks. The average position of hawks is attained using Eq. (2):

$$X_m(t) = \frac{1}{N} \sum_{i=1}^{N} X_i(t) \quad (2)$$

where $X_i(t)$ indicates the location of each hawk in iteration t and N denotes the total number of hawks. It is possible to obtain the average location in different ways, but we utilized the simplest rule.

2.2 Transition from Exploration to Exploitation

The HHO algorithm can transfer from exploration to exploitation then change between different exploitative behaviors based on the escaping energy of the prey. The escaping behavior of prey decreases the energy of prey considerably. The energy of prey is modeled as:

$$E = 2E_0(1 - \frac{t}{T}) \quad (3)$$

where E indicates the escaping energy of the prey, T is the maximum number of iterations, and E_0 is the initial state of its energy. In HHO, E_0 randomly changes inside the interval (-1, 1) at each iteration.

2.3 Exploitation Phase

According to the escaping behaviors of the prey and chasing strategies of the Harris' hawks, four possible strategies are proposed in the HHO to model the attacking stage. The preys always try to escape from threatening situations. Suppose that r is the chance of prey in successfully escaping (r <0.5) or not successfully escaping (r ≥0.5) before surprise pounce. Whatever the prey does, the hawks will perform a hard or soft besiege to catch the prey. In this regard, when $|E|$ ≥0.5, the soft besiege happens, and when $|E|$ <0.5, the hard besiege occurs.

- Soft besiege: when $r \geq 0.5$ and $|E| \geq 0.5$. This behavior is modeled by the following rules:

$$X(t+1) = \Delta X(t) - E \left| JX_{rabbit}(t) - X(t) \right| \quad (4)$$

$$\Delta X(t) = X_{rabbit}(t) - X(t) \quad (5)$$

where $\Delta X(t)$ is the difference between the position vector of the rabbit and the current location in iteration t, r_5 is a random number inside (0,1), and $J = 2(1 - r_5)$ represents the random jump strength of the rabbit throughout the escaping procedure. The J value changes randomly in each iteration to simulate the nature of rabbit motions.

- Hard besiege: when r ≥0.5 and $|E|$ <0.5. In this situation, the current positions are updated using Eq. (6):

$$X(t+1) = X_{rabbit}(t) - E \left| \Delta X(t) \right| \quad (6)$$

- Soft besiege with progressive rapid dives: when still $|E|$ ≥0.5 but r <0.5. This procedure is more intelligent than the previous case.

To mathematically model the escaping patterns of the prey, the levy flight (LF) concept is utilized in the HHO algorithm. To perform a soft besiege, the hawks can evaluate (decide) their next move based on the following rule in Eq. (7):

$$Y = X_{rabbit}(t) - E \left| JX_{rabbit}(t) - X(t) \right| \quad (7)$$

Then, they compare the possible result of such a movement to the previous dive to detect that will it be a good dive or not. If it was not reasonable, they also start to perform irregular, abrupt, and rapid dives when approaching the rabbit based on the LF-based patterns using the following rule:

$$Z = Y + S \times LF(D) \quad (8)$$

where D is the dimension of problem and S is a random vector by size $1 \times D$ and LF is the levy flight function, which is calculated using Eq. (9) (Yang, 2010):

$$LF(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}}, \sigma = \left(\frac{\Gamma(1+\beta) \times sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{(\frac{\beta-1}{2})}} \right)^{\frac{1}{\beta}} \quad (9)$$

where u, v are random values inside (0,1), β is a default constant set to 1.5. Hence, the final strategy for updating the positions of hawks in the soft besiege phase can be performed by Eq. (10):

$$X(t+1) = \begin{cases} Y & if F(Y) < F(X(t)) \\ Z & if F(Z) < F(X(t)) \end{cases} \quad (10)$$

where Y and Z are obtained using Eqs.(7) and (8).

- Hard besiege with progressive rapid dives
  When $|E|$ <0.5 and r <0.5, the following rule is performed in hard besiege condition:

$$X(t+1) = \begin{cases} Y & if F(Y) < F(X(t)) \\ Z & if F(Z) < F(X(t)) \end{cases} \quad (11)$$

where Y and Z are obtained using new rules in Eqs.(12) and (13).

$$Y = X_{rabbit}(t) - E\,|JX_{rabbit}(t) - X_m(t)| \quad (12)$$

$$Z = Y + S \times LF(D) \quad (13)$$

where $X_m(t)$ is obtained using Eq. (2).

The pseudocode of the proposed HHO algorithm is reported in Algorithm 1.

Algorithm 1: Pseudo-Code of HHO algorithm.

Inputs: The population size N and maximum number of iterations T
Outputs: The location of rabbit and its fitness value
Initialize the random population $X_i (i = 1,2,\ldots,N)$
while (stopping condition is not met) **do**
  Calculate the fitness values of hawks
  Set X_{rabbit} as the location of rabbit (best location)
  for (each hawk (X_i)) **do**
    Update the initial energy E_0 and jump strength J ▷ `E0=2rand()-1, J=2(1-rand())`
    Update the E using Eq. (3)
    if ($|E| \geq 1$) **then** ▷ `Exploration phase`
      Update the location vector using Eq. (1)
    if ($|E| < 1$) **then** ▷ `Exploitation phase`
      if ($r \geq$0.5 and $|E| \geq 0.5$) **then** ▷ Soft besiege
        Update the location vector using Eq. (4)
      else if ($r \geq$0.5 and $|E| < 0.5$) **then** ▷ Hard besiege
        Update the location vector using Eq. (6)
      else if ($r <$0.5 and $|E| \geq 0.5$) **then** ▷ Soft besiege with progressive rapid dives
        Update the location vector using Eq. (10)
      else if ($r <$0.5 and $|E| < 0.5$) **then** ▷ Hard besiege with progressive rapid dives
        Update the location vector using Eq. (11)
Return X_{rabbit}

3 HHO ALGORITHM FOR GENE SELECTION

The HHO algorithm was first used for solving the feature selection problem (FS) in (Thaher et al., 2020). The methodology used for converting the continuous HHO into binary is a two-step binarization method. In this method, the solution is transformed into binary but the real operators are used without converting. The solutions are converted into binary using two steps. The first step is by using a transfer function (TF) to convert the real solution R^n into intermediate vector $[0,1]^n$. Each element in the vector contains a probability of converting the corresponding feature or gene into either ”0” or ”1”.The second step is performed by using a specific binarization method.

This paper uses the S-shaped TF, which was used firstly in (Kennedy and Eberhart, 1997) to convert the continuous PSO into binary using Eq 14.

$$T(x_i^j(t)) = \frac{1}{1 + e^{-x_i^j(t)}} \quad (14)$$

4 PROPOSED BHHO BASED GENE SELECTION APPROACHES

This section describes the BHHO algorithm for GS with three fitness functions. The basic fitness function concerns the overall classification performance. The other new fitness functions are proposed to further increase the classification performance and minimize the number of selected genes in HHO GS framework.

4.1 Basic Fitness Function: Error Rate

The BHHO can be used to solve the GS problem using a single objective fitness function. This fitness

function utilizes the classification performance only. The main target is to maximize the classification accuracy or minimize the error rate in classification tasks. Eq 15 shows the basic fitness function which will be used as a baseline for comparison with the new fitness functions.

$$Fitness_1 = ErrorRate \tag{15}$$

Where $ErrorRate$ is determined according to Equation 6:

$$ErrorRate = \frac{FP+FN}{TP+TN+FP+FN} \tag{16}$$

Where TP, TN, FP and FN stand for true positives, true negatives, false positives and false negatives, respectively.

4.2 New Fitness Function: Error Rate and # Genes

Using the basic function in Eq 15, the BHHO may select a gene subset that has irrelevant or redundant genes. This is because the formula doesn't take into account minimizing the number of features, but focus on the classification performance only. This causes to select gene subsets that increases the classification performance and discard the number of selected genes. The problem of this approach is that the BHHO may select a gene subset with a classification performance that may be achieved by a smaller gene subset. To address this problem, a new multi-objective fitness function is proposed that aims to maximize the classification performance (minimize the classification error rate) and minimizes the number of genes. The formula of the new fitness function is shown in Eq 17.

$$Fitness_2 = \alpha_t \times \frac{\#Genes}{\#All\ Genes} + (1-\alpha_t) \times \frac{ErrorRate}{Error_0} \tag{17}$$

Where

$$\alpha_t = \alpha_{max} \times \frac{t}{T} \tag{18}$$

Where $\alpha \in [0,1]$. t denotes the tth iteration in the optimization process. $\#Genes$ represents the number of selected genes. $\#AllGenes$ stands for the number of all the available genes. $ErrorRate$ is the classification error rate got by the selected gene subset. $Error_0$ is the error rate got by using all the available genes. α_{max} is the predefined maximum value of α_t and $\alpha_{max} \in [0,1]$. T is the predefined maximum number of iterations for the BHHO evolutionary process.

Using Eq 17, the relative importance of the number of genes and the classification error rate is determined by the α_t and $(1-\alpha_t)$ respectively. The error rate is always assumed more significant than the size of the genes subset. Thus, the $(1-\alpha)$ is assigned to value greater than the α_{max}. α_t increases linearly throughout the optimization process of the BHHO. This means that the error rate has greater importance at the initial stages of the optimization process. The linear increment of the α_t causes to reduce the importance of the error rate at the late stages. The opposite case occurs for the number of selected genes. The size of a genes subset is given larger weights at the latter stages. Another matter is that the number of genes is much larger than the error rate. Thus, a normalization step is required to balance these components. This is done by dividing the size of genes subset by the total number of genes. The result will fall in the range (0,1]. Besides, the classification error rate is divided by the error rate got by using all available genes. This will transform the error rate to a value in the range [0,1]. Normalizing the error rate is a necessary step because in some microarray data sets, the error rate changes in a small range throughout the evolutionary process. Another issue is that $\frac{ErrorRate}{Error0}$ may be larger than 1 at the begging of the optimization process. This is normal because the error rate is required to dominate the evolutionary process at the initial stages. When α_t increases to a relatively large value, BHHO reduces the value of $ErrorRate$ to be smaller than the $Error_0$.

4.3 New Fitness Function: A Two-stage Approach

In the proposed $Fitness_2$, using a linear increasing weight can balance the error rate and the number of selected genes. Thus, it can solve the problem of selecting redundant and irrelevant genes in the gene subset. However, there is still a problem that the BHHO selects a small gene subset with low classification performance instead of selecting a large gene subset with high classification performance. To solve this problem, $Fitness_3$ is proposed to perform GS in two stages. This implies that the entire optimization process is divided into two stages. In the first stage, the BHHO concerns optimizing the classification performance. In the second stage, the number of selected genes is considered in the fitness function. In the two-stage fitness function, the BHHO uses the solutions found in the first stage to perform further optimization for the number of genes. This causes the BHHO to start in the second stage with optimized solutions that achieved suitable classification performance in the first stage. The fitness function used in this two-stage GS approach is shown in Eq 19.

$$Fitness_3 = \begin{cases} ErrorRate, & \text{Stage1} \\ \alpha \times \frac{\#Genes}{\#All\ Genes} + (1-\alpha) \times \frac{ErrorRate}{Error_0}, & \text{Stage2} \end{cases} \quad (19)$$

Where α is constant values and $\alpha in [0,1]$. α shows the relative importance of the number of features and $(1-\alpha)$ shows the relative importance of the classification error rate. $ErrorRate$, $\#Genes$, $\#All\ Genes$, $ErrorRate$, $Error_0$ are the same as the ones used in $Fitness_2$. As the classification performance is assumed to be more important than the number of features, α is set to be smaller than $(1-\alpha)$.

5 EXPERIMENTAL DESIGN

All the experiments were executed on a personal machine with AMD Athlon Dual-Core QL-60 CPU at 1.90 GHz and memory of 2 GB running Windows7 Ultimate 64 bit operating system. The optimization algorithms have been all implemented in Python in the EvoloPy-FS framework (Khurma et al., 2020). Ten microarray datasets taken from http://csse.szu.edu.cn/staff/zhuzx/Datasets.html are used to evaluate the performance of the proposed approach. The selected benchmark data sets are commonly used in many studies and cover the example of small, medium, and large dimensional data sets. Table 1 shows the characteristics of the selected datasets.

Table 1: Gene expression datasets characteristics.

NO	Datasets	# Genes	# Samples	# classes
1	Breast	24,481	97	2
2	MLL	12,582	72	3
3	Colon	2000	62	2
4	ALL-AML	7129	72	2
5	ALL-AML-3C	7129	72	3
6	ALL-AML-4C	7129	72	4
7	CNS	7129	60	2
8	Ovarian	15,154	253	2
9	SRBCT	2308	83	4
10	Lymphoma	4026	62	3

The maximum number of iterations and the population size were set to 100 and 10 respectively. In this work, the K-nn classifier (K = 5) is used to assess the goodness of each solution in the wrapper GS approach. Each data set is randomly divided into two parts; 80% for training and 20% for testing. To obtain statistically significant results, this division was repeated 30 independent times. Therefore, the final statistical results represent the average over 30 independent runs. As the maximum iteration is 100, in the two-stage approach, the first 50 iterations are the first stage and the last 50 iterations are the second stage. We assume the number of genes is important in GS but much less important than classification accuracy. Therefore, $\alpha_{max} = 0.2$ in Eq 17 and $\alpha = 0.2$ in Eq 19 in the second stage of the two-stage approach. BGWO, BCS, and BBA were used for comparison with the proposed approaches. The parameters settings of them as follows: in GWO α value is [2,0]. In BA, Qmin Frequency minimum is 0, Qmax Frequency maximum is 2, A Loudness is 0.5, r Pulse rate is 0.5. In CS, pa value is 0.25 and β is 3/2.

The proposed evaluation measures are classification accuracy and number of selected genes.

6 RESULTS AND DISCUSSIONS

The experimental results of the three approaches on ten datasets are shown in Table 2. In the table, "All" means that all of the available genes are used for classification. BHHO-Er stands for the BHHO based GS approach with Eq 15 as the fitness function. BHHO-ErNo and BHHO-2Stage represent the two proposed GS approaches with Eq 17 and Eq 19 as fitness functions, respectively. "#A" and "#G" show the average test accuracy and the size of the gene subsets selected by each algorithm in 30 runs respectively. "Std-Acc" represents the standard deviation of the 30 test accuracy achieved by each algorithm.

6.1 Results of BHHO with Basic Fitness Function

Inspecting Table 2, it can be seen that the BHHO-Er algorithm can select gene subsets with a sufficient number of genes and high classification performance. In comparison with the results of BHHO with all the gene subsets, BHHO-Er achieves higher classification accuracy in almost all microarray data sets. Regarding the number of selected genes, it appears that the BHHO-Er minimized the number of selected genes to half of the original number of genes. The 5K-nn classifier with BHHO-Er obtained a classification accuracy similar to using all genes across only one data set which is the Breast data set. The results suggest that BHHO-Er can efficiently select a subset of relevant genes that contains around half of the original genes and increase the classification performance. For example, in the ALL-AML4c data set, with all the 7,129 genes, 5K-nn could achieve a classification accuracy of 81.425% while with 3,427 genes, it can increase the classification accuracy to 83.569%. All the standard deviation values shown by "StdAcc" are smaller than 0.05 which indicates the stability of the proposed approach.

Table 2: Experimental results.

Algorithm		Dataset				
		Breast	MLL	Colon	ALL-AML	ALL-AML3c
All	#A	75.800	94.000	78.500	85.100	82.211
	#G	24,481	12,582	2000	7129	7129
BHHO-Er	#A	75.800	94.420	79.256	85.742	83.006
	#G	12,175.217	8,052.48	1,031.111	3,831.838	2,176.441
	#A-Std	2.111E-17	1.755E-3	52.800E-5	1.359E-3	4.251E-2
BHHO-ErNo	#A	75.800	95.341	79.391	85.822	83.140
	#G	6,242.655	7,839.554	853.333	2,815.955	1,792.734
	#A-Std	2.111E-17	1.536E-3	50.211E-5	1.729E-3	4.671E-2
BHHO-2Stage	#A	75.800	95.958	79.455	85.963	83.620
	#G	5,451.103	4,936.015	811.111	2,560.499	1,866.121
	#A-Std	2.111E-17	2.189E-3	44.369E-5	1.952E-5	5.008E-2
		ALL-AML4c	CNS	Ovarian	SRBCT	Lymphoma
All	#A	81.425	66.456	91.623	87.623	98.315
	#G	7,129	7,129	15,154	2308	1026
BHHO-Er	#A	83.569	67.889	94.569	87.612	98.450
	#G	3,427	3,389.840	7,631.773	1,129.443	504.087
	#A-Std	1.430E-2	4.223E-2	1.456E-2	2.854E-2	17.857E-5
BHHO-ErNo	#A	84.001	68.201	95.226	87.655	98.596
	#G	3,427.012	2,691.198	7,248.359	1,104.516	493.910
	#A-Std	1.375E-2	4.985E-2	1.369E-2	2.159E-2	7.589E-5
BHHO-2Stage	#A	84.250	68.025	95.631	87.598	98.629
	#G	2,828.685	2,644.859	5,720.635	1,114.0716	503.106
	#A-Std	1.231E-2	5.079E-2	1.175E-2	2.006E-2	8.438E-5

6.2 Results of BHHO with New Fitness Function: Error Rate and #Genes

Based on the results in Table 2, the selected gene subsets in most of the cases contains fewer than half of the original number of genes. The highest reduction rate was achieved on the Breast data set, which is around 75% of the original genes. In comparison with the accuracy obtained by all the gene sets, the BHHO-ErNo got a higher classification rate across all the data sets except the Breast data set. BHHO-ErNo can search in the gene space and obtain gene subsets with a smaller size than those obtained by the BHHO-Er across all the data sets. The reduction rate across four data sets is more than 16%. The highest reduction rates were around 49% on the Breast data set and 27% on the ALL-AML data set. Regarding the standard deviation results, there is no clear difference between the BHHO-Er and the BHHO-ErNo. This supports the idea that adding the number of genes to the formula of fitness function as in Eq 17 helps the BHHO to search in the gene space for smaller size gene subsets. The removal of the irrelevant and redundant genes can simplify the generated learning model and achieve better classification results on testing.

6.3 Results of BHHO with New Two-Stage Fitness Function

Based on Table 2, the results of BHHO-2Stage shows that the size of the selected gene subsets using this approach reached to around 60% of the number of original genes in many cases. The Breast data set achieved the highest reduction rate of around 78%. The generated gene subsets by BHHO-2Stage achieved higher classification performance compared with using all genes across all the data sets except the Breast data set. By comparing BHHO-2Stage with BHHO-Er, BHHO-2Stage can generate smaller size gene subsets than BHHO-Er. In comparison with the BHHO-Er, the reduction of the average size is more than 20% in six of ten data sets and it is around 55% in the Breast data set. With smaller gene subsets, BHHO-2Stage achieves better or same classification performance as "BPSO-Er" in almost all data sets. The standard deviation values in the two approaches are very close.

7 COMPARISONS WITH OTHER WRAPPER-BASED GS METHODS

To show the performance of the proposed fitness functions, we compare the best-proposed approach, which is BHHO-2Stage with three well-known wrapper-based GS methods. These are BGWO, BBA, and BCS. We will apply theses GS approaches with a 2 stage fitness function. Taking the Breast data set as an example, BGWO selects 9,559 genes with a classification performance 72.44%, BCS selects 7,360 genes with classification performance 74.84%, and BBA selects 6,289 genes with a classification performance of 64.23%. The average number of genes in BHHO-2Stage is 5,451.103 with classification performance 75.800. These results show the superiority of the

BHHO-2Stage approach to other well-known GS approaches using the same fitness function.

8 CONCLUSIONS

This paper presents a new GS method based on the BHHO algorithm and KNN classifier. Two new fitness functions are proposed. The first fitness function combines the classification performance and the size of the genes subset in one formula. A linear weight is used to balance these components together. The second one is a two-stage fitness function. This focuses on optimizing the classification performance in the first stage and optimizing the number of genes in the second stage. The new two fitness functions were compared with a common fitness function that uses the classification performance only in a BHHO based-wrapper GS approach. The results show that BHHO with the fitness function that uses the classification performance only can improve the classification performance using all genes. In almost all the data sets, BHHO with either of the two proposed fitness functions could achieve higher classification performance. Besides, they could achieve a fewer number of genes than BHHO with overall classification performance as the fitness function. BHHO with the two-stage fitness function outperforms the linearly changing weights fitness function in most problems based on the classification performance and the number of genes selected. BHHO with the proposed fitness functions can successfully reduce the number of genes and achieve higher classification performance. In the future, we will investigate a BHHO-based evolutionary multi-objective GS approach to explore the Pareto front of non-dominated solutions.

ACKNOWLEDGMENTS

This work is supported by the Ministerio español de Economía y Competitividad under project TIN2017-85727-C4-2-P (UGR-DeepBio).

REFERENCES

Al-Betar, M. A., Alomari, O. A., and Abu-Romman, S. M. (2020). A triz-inspired bat algorithm for gene selection in cancer classification. *Genomics*, 112(1):114–126.

Alomari, O. A., Khader, A. T., Al-Betar, M. A., and Alyasseri, Z. A. A. (2018). A hybrid filter-wrapper gene selection method for cancer classification. In *2018 2nd International Conference on BioSignal Analysis, Processing and Systems (ICBAPS)*, pages 113–118. IEEE.

Chuang, L.-Y., Yang, C.-S., Wu, K.-C., and Yang, C.-H. (2011). Gene selection and classification using taguchi chaotic binary particle swarm optimization. *Expert Systems with Applications*, 38(10):13367–13377.

Emary, E., Zawbaa, H. M., and Hassanien, A. E. (2016). Binary grey wolf optimization approaches for feature selection. *Neurocomputing*, 172:371–381.

Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., and Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future generation computer systems*, 97:849–872.

Kennedy, J. and Eberhart, R. C. (1997). A discrete binary version of the particle swarm algorithm. In *1997 IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation*, volume 5, pages 4104–4108. IEEE.

Khurma., R. A., Aljarah., I., and Sharieh., A. (2020). An Efficient Moth Flame Optimization Algorithm using Chaotic Maps for Feature Selection in the Medical Applications. In *Proceedings of the 9th International Conference on Pattern Recognition Applications and Methods - Volume 1: ICPRAM,*, pages 175–182. INSTICC, SciTePress.

Khurma, R. A., Aljarah, I., Sharieh, A., and Mirjalili, S. (2020). Evolopy-fs: An open-source nature-inspired optimization framework in python for feature selection. In *Evolutionary Machine Learning Techniques*, pages 131–173. Springer.

Moghadasian, M. and Hosseini, S. P. (2014). Binary cuckoo optimization algorithm for feature selection in high-dimensional datasets. In *International conference on innovative engineering technologies (ICIET'2014)*, pages 18–21.

Mohamad, M. S., Omatu, S., Deris, S., and Yoshioka, M. (2011). A modified binary particle swarm optimization for selecting the small subset of informative genes from gene expression data. *IEEE Transactions on information Technology in Biomedicine*, 15(6):813–822.

Thaher, T., Heidari, A. A., Mafarja, M., Dong, J. S., and Mirjalili, S. (2020). Binary harris hawks optimizer for high-dimensional, low sample size feature selection. In *Evolutionary Machine Learning Techniques*, pages 251–272. Springer.

Tran, B., Xue, B., and Zhang, M. (2014). Improved pso for feature selection on high-dimensional datasets. In *Asia-Pacific Conference on Simulated Evolution and Learning*, pages 503–515. Springer.

Xi, M., Sun, J., Liu, L., Fan, F., and Wu, X. (2016). Cancer feature selection and classification using a binary quantum-behaved particle swarm optimization and support vector machine. *Computational and mathematical Methods in Medicine*, 2016.

Yang, X.-S. (2010). *Nature-inspired metaheuristic algorithms*. Luniver press.

Zawbaa, H. M., Emary, E., Grosan, C., and Snasel, V. (2018). Large-dimensionality small-instance set feature selection: A hybrid bio-inspired heuristic approach. *Swarm and Evolutionary Computation*, 42:29–42.