ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY
Volume 26, Number 3–4, 2023, 351–364

Binary Anarchic Society Optimization for Feature Selection

Umit KILIC[1, *], Esra SARAC ESSIZ[1], and Mumine KAYA KELES[1]

[1]Department of Computer Engineering, Adana Alparslan Turkes Science and Technology University, Saricam, 01230, Adana, Turkey

Email: ukilic@atu.edu.tr*, esarac@atu.edu.tr, mkaya@atu.edu.tr

* Corresponding author

Abstract. Datasets comprise a collection of features; however, not all of these features may be necessary. Feature selection is the process of identifying the most relevant features while eliminating redundant or irrelevant ones. To be effective, feature selection should improve classification performance while reducing the number of features. Existing algorithms can be adapted and modified into feature selectors. In this study, we introduce the implementation of the Anarchic Society Optimization algorithm, a human-inspired algorithm, as a feature selector. This is the first study that utilizes the binary version of the algorithm for feature selection. The proposed Binary Anarchic Society Algorithm is evaluated on nine datasets and compared to three known algorithms: Binary Genetic Algorithm, Binary Particle Swarm Optimization, and Binary Gray Wolf Optimization. Additionally, four traditional feature selection techniques (Info Gain, Gain Ratio, Chi-square, and ReliefF) are incorporated for performance comparison. Our experiments highlight the competitive nature of the proposed method, suggesting its potential as a valuable addition to existing feature selection techniques.

Key-words: artificial intelligence; anarchic society optimization; feature selection; meta-heuristic; swarm intelligence algorithms.

1. Introduction

Data mining has become a rapidly growing research field driven by the need to extract valuable information from large datasets. This massive data has sparked studies across various fields including verification [1], clustering [2], modeling [3], control systems [4], neural networks [5], and health [6]. Data mining is a part of the knowledge discovery process, encompassing data preprocessing, mining, pattern evaluation, and knowledge presentation [7]. Dealing with relevant features in data is crucial for achieving optimal performance and reducing computational time. This can be achieved by feature selection (FS). FS is the process of removing irrelevant features from a dataset and can be categorized as wrapper or filter techniques [8]. Discovering an optimal feature subset through the search process is a challenging problem due to the exponential growth of the search space with increasing feature count. In practice, this can be restrictive and costly [9].

The use of metaheuristic algorithms has gained attention in recent years as a way to address the challenge of finding the best feature subset. While these algorithms do not guarantee the best solution in every run, they are able to detect optimal solutions within a reasonable time frame [10]. Researchers have proposed several nature-inspired metaheuristic algorithms, including insects [11–13], reptiles [14, 15], birds [16–18], animals [19–22], sea creatures [23, 24], plants [25, 26], human-based algorithms [27–29], for solving optimization problems. Some of these algorithms have been modified to become feature selectors. These algorithms are discussed in Section 2. The purpose and motivation of FS are to remove irrelevant and/or redundant features without affecting learning performance and to keep only relevant features. Many algorithms are emerging in the field of FS, where new metaheuristic methods are proposed every day. In this study, a modification of the Anarchic Society Optimization (ASO) algorithm, a human-inspired metaheuristic algorithm, is proposed to be used as a feature selector. Its binary version, Binary Anarchic Society Optimization (BASO), is introduced for FS. The performance of BASO is evaluated on several UCI (http://archive.ics.uci.edu/ml) datasets commonly used in machine learning research.

ASO algorithm, introduced by Ahmadi-Javid in 2011, is a nature-inspired swarm intelligence algorithm that simulates human society [27]. In this algorithm, potential solutions are represented as members of an anarchic society that struggle to make their situation better by behaving anarchically. The details of the ASO algorithm are described in Section 3. The results of the BASO algorithm are compared with conventional FS algorithms such as Info Gain (IG), Gain Ratio (GR), Chi-square (CHI), and ReliefF. Besides, the BASO results are compared with state-of-the-art FS algorithms including Binary Grey Wolf Optimization (BGWO) [30], Binary Particle Swarm Optimization (BPSO) [31] and Binary Genetic Algorithm (BGA) [32]. Genetic Algorithm (GA), proposed by Holland [33], is an optimization and search algorithm inspired by natural selection. Particle Swarm Optimization (PSO) is a swarm intelligence optmization algorithm proposed by Kennedy and Eberhart [16]. It addresses the problems by creating a population that represents the candidate solutions. It solves the problems by creating a population that represents the candidate's solutions. Grey wolf optimization (GWO) algorithm, proposed by Mirjalili et al. [19], is another swarm intelligence optimization algorithm that emulates the hunting behavior of wolves.

The main contributions of this paper can be summarized as follows:

- The BASO algorithm proposed in this study was utilized as an FS method for the first time.
- In order to evaluate the performance of the proposed algorithm, four optimization algorithms, including the proposed method, were tested on a set of benchmark datasets including low, medium, and high dimensional datasets.
- The proposed algorithm was compared to well-established traditional FS methods, such as IG, GR, CHI, and ReliefF, commonly used in the literature, to demonstrate its success and superiority.
- A distinguishing aspect of BASO is its consideration of the past in addition to personal best and global best solutions.
- Compared to other algorithms, BASO exhibited a greater ability to extract significant historical correlations from medium or large datasets compared to small datasets.

- According to the experimental results, it is concluded that the BASO algorithm performs better in terms of F-measure and the number of selected features.
- This study indicates that not following the best solution and allowing the herd to become anarchic, adventurous, and indecisive can yield good results.

The paper is organized as follows: a literature review is provided in Section 2. The details of the ASO algorithm are outlined in Section 3. The proposed binary version of ASO, named BASO, is described in Section 4. The performance evaluation of BASO and comparisons with other techniques are presented and discussed in Section 5. Conclusions are in Section 6.

2. Literature Review

High dimensionality in the feature space presents a challenge, and feature selection has been proposed as a solution to overcome this issue [34]. Traditional FS strategies such as CHI and IG, which utilize feature scoring measures, have limitations in that they do not interact with the classifier model, leading to decreased classification efficiency. In contrast, wrapper methods that incorporate classification methods in the searching process have been proposed to achieve better performance [9]. Metaheuristic methods have been adapted for the purpose of FS by utilizing a wrapper method that incorporates classifiers. Taradeh et al. [35] employed Gravitational Search Algorithm (GSA) as a feature selector and used K-Nearest Neighbors (KNN) and Decision Tree (DT) classifiers as evaluators. The proposed method was assessed on 18 UCI datasets using GA, PSO, and GWO. The results demonstrate the superiority of the proposed method in the FS process.

The Artificial Bee Colony (ABC) algorithm is widely used in the literature for FS in the nature-inspired category. Palanisamy and Kanmani [8] proposed a new technique using ABC that simulates honey bee foraging behavior. They applied ABC for FS and compared it with J48, Bagging, Boosting, and Ant Colony Optimization-based FS algorithms on ten UCI datasets. Their method achieved superior results. Another study by Zorarpaci and Ozel [9] presented a hybrid approach combining ABC and Differential Evolution (DE) algorithms for FS. They compared it with pure ABC, DE, and popular FS methods such as correlation-based FS, CHI, and IG on fifteen UCI datasets. The results showed that the hybrid approach outperformed other methods for most datasets. ABC is employed as a feature selector by Kilic and Kaya Keles to see its performance on medical datasets [36–38]. Moreover, Kaya Keles et al. also used binary ABC (BABC) as a feature selector to predict concrete strength [39] and to anticipate the leadership perception of site managers [40].

Mafarja and Mirjalili [41] introduced two hybrid methods that combine the whale optimization algorithm (WOA) with simulated annealing (SA) for FS. The first method embeds SA into WOA, while the second method employs SA to improve exploitation after each iteration of WOA. The performance of these methods is evaluated using 18 standard benchmark datasets from the UCI repository, and the measured performance is compared against native WOA, ant lion optimizer (ALO), genetic algorithm (GA), and particle swarm optimization (PSO). The results demonstrate the efficiency of the proposed methods in improving classification accuracy. Another nature-inspired optimization algorithm, the Butterfly Optimization Algorithm, is converted into a feature selector by Arora and Anand [42]. The continuous form is turned into a binary form using V-shaped and S-shaped transfer functions. 21 well-known UCI datasets are used for the assessment and proposed approaches are compared to state-of-the-art FS methods such as

GA, PSO, ALO, Brain Strom Optimizer, GWO, Dragonfly Algorithm (DA), Sine-Cosine Algorithm (SCA), SSA and WOA. Better results than other approaches are obtained by the proposed approaches for FS, according to the experiments.

The black widow optimization algorithm, inspired by the mating behavior of black widow spiders, was proposed by Hayyolalam and Kazem [22]. This proposed solution for nonlinear optimization problems is turned into a feature selector in [43]. They used 20 benchmark datasets. A comparison with BPSO, V-shaped PSO, and BGWO was made. The results show the efficiency of the proposed method. Mafarja et al. [44] proposed a hybrid metaheuristic method employing GWO and WOA to reduce the drawbacks of both algorithms. The developed wrapper-based FS method is used on 18 well-known datasets of the UCI repository. Comparisons are done using four state-of-the-art methods which are binary GOA, binary GSA, GA, and BPSO. Their results showed that the proposed method surpasses state-of-the-art approaches, notably. Another hybridization is made by Qaraad et al. [45] to deal with the shortcomings of algorithms. To enhance the performance of SSA, the GWO strategy is included in SSA. The position of leaders in SSA is improved by applying robust exploitation of SSA. Furthermore, GWO's exploration strategy is included to update the followers' positions. By making those changes, the exploration and exploitation phases are expected to show improvements. 18 datasets are used to see the effectiveness of the proposed method. Comparisons are done with original SSA, GWO, and some other known methods such as WOA, SCA, and so on. The outputs show the efficiency of the proposed algorithm and the improvement in the exploitation and exploration phases. In the paper from Hu et al. [46], a multi-surrogate assisted BPSO was proposed and experimented on the UCI datasets. According to the results, the proposed method was efficient in reducing running time and prediction error.

3. Overview of the Anarchic Society Optimization

Anarchic Society Optimization (ASO), proposed by Ahmadi-Javid, is a swarm intelligence algorithm that mimics the behavior of an anarchic social group. It incorporates fickle, venturesome, and frequently irrational behavior to resist stability. This approach promotes a high level of diversity among individuals and thorough exploration of the solution space, thereby avoiding local optima. Each individual in ASO represents a potential solution to the optimization problem. According to the author, ASO shows promise in effectively exploring solution spaces and preventing the algorithm from getting trapped in local optima. In the literature, ASO has been applied to solve various problems, including scheduling problems [47–49] for control of automatic voltage regulators [50], to optimize the operation of reservoir systems [51] and for water distribution networks [52].

The algorithm is based on an anarchic social group whose members behave abnormally to improve their state. The process starts with the random initialization of members. All members are informed of the best position in the first *k* iteration, represented as *GBest(k)*, which is the global best. Members are also aware of the best member's position in iteration *k*, denoted by $i^*(k)$. The best personal position of members is stored, and $P_i(k)$ represents the best personal position of the ith member in iteration k. $X_i(k)$ denotes the position of the ith member in the kth iteration.

Following random initialization, the algorithm introduces three movement policies (MPs) for each member to update their position by combining these policies, specifically the MP current, MP society, and MP past. Three index values are also presented, including the fickleness index

(*FI*), external irregularity index (*EI*), and internal irregularity index (*II*). These indexes are employed in the calculations of the movement policies. The first policy, denoted as $MP_i^{Current}(k)$, is based on the current position of the ith member in iteration k. With this movement policy, members can choose different or common neighboring methods. In this MP, the *FI* is employed, which represents the members' indecision level compared to others' situations and is defined as follows:

$$FI_i(k) = \frac{f(X_i(k)) - f(X_{i^*(k)}(k))}{f(X_{i^*(k)}(k))}, \tag{1}$$

where $FI_i(k)$ represents the fickleness index for ith member in kth iteration. The second movement policy, represented as $MP_i^{Society}(k)$, in the ASO algorithm is based on the positions of other members. Although logically, members would generate their policy based on the global best or the best member in a related iteration, the anarchic behavior of the ASO algorithm allows members to choose their movement policy based on any (or a number) of other members. In this MP, the EI is used. EI represents the level of diversity and inequality among the members of the society. As the diversity of society increases, members become more anxious and tend to move in an anarchical manner. The formula to calculate $EI_i(k)$ is

$$EI_i(k) = \frac{max_j\{f(X_i(k))\} - f(X_{i^*(k)}(k))}{f(X_{i^*(k)}(k))}, \tag{2}$$

where $EI_i(k)$ symbolizes the external irregularity index for ith member in kth iteration. For the minimization problem, *max* indicates the worst member and *j* is its index where *j* is not equal to i. The third and final movement policy is based on past positions and is represented as $MP_i^{Past}(k)$. Normally, members would produce a movement policy based on their best previously visited position, which is called their personal best. At this point, the anarchic nature of the ASO algorithm takes over, and members may choose any (or a number of) past positions to generate the policy. In this MP,II, which evaluates the past situations experienced by each member, is used. When a member's best personally experienced situation differs significantly from the global best situation, the member cannot concentrate on their next movement and tends to behave more erratically. $II_i(k)$ is calculated as follows [48]:

$$II_i(k) = \frac{f(P_i(k)) - f(GBest(k))}{f(GBest(k))}, \tag{3}$$

where $II_i(k)$ stands for the ith member's internal irregularity index in the kth iteration. After calculating these values, the movement policies should be taken into consideration. The values for $MP_i^{Current}(k)$, $MP_i^{Society}(k)$, and $MP_i^{Past}(k)$ are defined in (4), (5), and (6), respectively:

$$MP_i^C(k) = \left\{ \begin{array}{ll} \text{Move along a direction randomly} & \\ \text{chosen from the hypercube} & i \neq i^*(k), \\ [(1-\epsilon)xDir_i(k-1), (1+\epsilon)xDir_i(k-1)] & FI_i(k) \leq \beta_1 \\ \text{with random stepsize,} & \\ & \\ \text{Move toward the situation of a randomly selected} & i \neq i^*(k), \\ \text{member with random stepsize,} & FI_i(k) > \beta_1 \\ & \\ \text{Move toward the situation } CO_{0,1}(X_i(k), GBest(k)), & i = i^* \end{array} \right\}, \quad (4)$$

$$MP_i^S(k) = \left\{ \begin{array}{ll} \text{Move toward the situation of the best member } i^*(k) & \\ \text{with random stepsize,} & EI_i(k) \leq \beta_2 \\ & \\ \text{Move toward the situation of a randomly selected} & \\ \text{member with random stepsize,} & EI_i(k) > \beta_2 \end{array} \right\}, \quad (5)$$

$$MP_i^P(k) = \left\{ \begin{array}{ll} \text{Move toward the best situation previously experienced} & \\ \text{with random stepsize,} & II_i(k) \leq \beta_3 \\ & \\ \text{Move toward the situation of a randomly selected} & \\ \text{member with random stepsize,} & II_i(k) > \beta_3 \end{array} \right\}. \quad (6)$$

In the definitions of movement policies, $\beta_1, \beta_2, \beta_3$, and ϵ are positive constants, $CO_\alpha(X, Y)$ is a blended crossover, $Dir_i(k)$ is movement direction, which is defined as follows for each member i in iteration k [49]:

$$Dir_i(k) = X_i(k+1) - X_i(k). \quad (7)$$

The combination of movement policies in ASO can be sequential, elitism, or hybridization, as explained in [49]. Figure 1 given in [56] illustrates the general framework of the ASO algorithm. While ASO shares similarities with PSO, it introduces an additional criterion based on the past situation of members. Unlike PSO, designed for continuous problems, ASO is capable of addressing both continuous and discrete problems.

4. The Proposed Binary Anarchic Society Optimization

In the ASO algorithm, a member's position represents a potential solution for the FS problem. For FS, fewer features that yield higher accuracy are preferable. To enable an optimization algorithm to select features, modifications must be made. In the case of wrapper methods, the binary version of the algorithm needs a fitness value producer to evaluate the position of the solution members.

The proposed BASO utilizes the KNN algorithm to evaluate the members' positions and produce fitness values. KNN is chosen due to its frequent usage in literature. Initially, the algorithm generates random positions for the members and evaluates each solution using the F-measure. The member with the best feature subset is marked as i^* and $GBest$. The main loop then begins, where in each iteration, the *FI*, *EI*, and *II* values are calculated for each member using (8), (9), and (10), respectively:

$$FI_i(k) = \frac{f(X_i(k)) - f(X_{ib}(k))}{f(X_{ib}(k))}, \tag{8}$$

where the fitness function is denoted by f, and $X_i(k)$ represents the position of the i-th member in the k-th iteration. Similarly, $X_{ib}(k)$ denotes the position of the best member in the k-th iteration,

$$EI_i(k) = \frac{f(X_w(k)) - f(X_{ib}(k))}{f(X_{ib}(k))}, \tag{9}$$

where $X_w(k)$ represents the position of the member that has the worst fitness value in kth iteration,

$$II_i(k) = \frac{f(Pbest_i(k)) - f(GBest(k))}{f(GBest(k))}. \tag{10}$$

where $Pbest_i(k)$ symbolizes position of personal best of ith member and $GBest(k)$ denotes position of the global best member in iteration k.

β_1, β_2, and β_3 are positive constants in the original algorithm. In the proposed BASO, these values are set up as the mean of *FI*, *EI*, and *II*, in order. Based on our experiments, it was found that employing a single pre-specified value for each constant is inadequate and can lead to poor performance, owing to the diversity and dissimilarity of the datasets used. Therefore, in the proposed method, each movement policy is applied to the members individually. Subsequently, an elitism rule is applied to select the best-produced position based on the F-measure value among the three-movement policies. The chosen position is then assigned to the corresponding member. The best member of the iteration and global best are identified and stored. The subsequent iteration begins and this cycle repeats until the termination criteria are satisfied. The movement policies, modified for the FS problems, are shown in (11), (12), and (13), respectively:

$$MP_i^{Current}(k) = \left\{ \begin{array}{ll} \text{Create a random member and apply} & i \neq i^*(k), \\ \text{a single point (SP) crossover with it} & FI_i(k) \leq \beta_1 \\ & \\ \text{Apply SP crossover with a random} & i \neq i^*(k), \\ \text{number } j \text{ where } j \neq i, & FI_i(k) > \beta_1 \\ & \\ \text{Apply SP crossover with the } GBest, & i = i^* \end{array} \right\}, \tag{11}$$

$$MP_i^{Society}(k) = \left\{ \begin{array}{ll} \text{SP crossover with the best member} & \\ \text{of the iteration } (i^*(k)), & EI_i(k) \leq \beta_2 \\ & \\ \text{SP crossover with a random} & \\ \text{member j } (j \neq i), & EI_i(k) > \beta_2 \end{array} \right\}, \tag{12}$$

$$MP_i^{Past}(k) = \left\{ \begin{array}{ll} \text{SP crossover with the personal best position,} & II_i(k) \leq \beta_3 \\ \text{SP crossover with a random member j } (j \neq i), & II_i(k) > \beta_3 \end{array} \right\}. \quad (13)$$

$Dir_i(k)$ and ϵ are not needed to be used in the proposed BASO. Pseudo-code of the proposed BASO can be seen in Algorithm 1 given in [56]. Figure 2 given in [56] provides an overview of the FS process employed in this study. Initially, the original datasets are input to the proposed method, which evaluates the goodness of subsets to determine whether the feature subset is an improvement. The termination criteria are then checked at the end of each iteration.

5. Experimental Results and Discussion

The following pages provide details for experimental configurations, results, and discussions.

5.1. Experimental setup

The proposed BASO algorithm was implemented in MATLAB and evaluated on nine UCI datasets. KNN was used as part of BASO to calculate the F-measure of feature subsets. The experiments were conducted on a personal computer with an Intel Core i5-6400 CPU and 16 GB RAM. BASO's performance was compared to three other popular algorithms (BGA, BPSO, and BGWO) converted into feature selectors from the literature. The population size and maximum iteration number were set to 10 and 100, respectively, for all algorithms. There are various methods available for validating a classifier, including cross-validation, confusion matrix analysis, ROC curve analysis, and Cohen's kappa score. For the purpose of validating our model, we have opted to utilize the cross-validation method [55]. 10-fold cross-validation is employed for the experiments. Similar architecture and the same parameters are used for all algorithms. The parameters such as k (in KNN), population number, maximum iteration, and fold values are defined based on common usage in the literature [30,31,53,54]. The comparisons are made utilizing the F-measure value and the average number of the selected features. No objective function was employed, and each experiment was conducted independently for each dataset. The datasets are described in Table 1, and the parameter settings are given in Table 2.

Table 1. List of the datasets used in this work

Datasets	No. of attributes	No. of samples
Breast-w	9	699
Car	6	1728
Credit-g	20	1000
Dermatology	34	366
Glass	9	214
Heart-c	13	303
Ionosphere	34	351
Lymphography	18	148
Sonar	60	208

Table 2. Parameter settings for the experiments

Parameter	Value(s)
K for KNN	3
Dimension of population	10
Number of iterations	100
Number of runs	10
n (cross-validation)	10
Problem dimension	Number of features in dataset
Acceleration constants in PSO	[2,2]
Inertia w in BPSO	[0.9,0.4]
Parameter A in BGWO	min=0 max=2

5.2. Results and discussions for the datasets

To establish a benchmark, the F-measure and the number of selected features were recorded for each run and averaged over ten iterations. Table 3 presents the resulting average values, with the best results highlighted in bold.

The analysis of the table shows that BASO performed well on datasets with a large number of features and samples. However, its performance was relatively poorer when the number of samples increased without a corresponding increase in features. However, despite the small differences between the results, the number of selected features can still play an important role in selecting the best method. Based on the results presented in Table 3, BASO outperforms BGA, BPSO, and BGWO in terms of F-measure for the glass, heart-c, lymphography, and sonar datasets. For the ionosphere dataset, BASO and BGA achieve the same level of performance, sharing first place in terms of F-measure and the number of selected features. However, for the breast-w, car, and credit-g datasets, BGA obtains slightly better results than BASO. For the dermatology dataset, BGWO achieves the best score, but BASO, BGA, and BPSO obtain very similar results with fewer features. BGWO achieves a score of 0.989 using 25.6 features, while BASO and BGA achieve a score of 0.988 with 19.7 and 19.4 features, respectively. This represents a significant reduction of 23% and 24.2% in the number of features, while the F-measure value only decreases by 0.1%. This drastic reduction in the number of features justifies accepting a slight decrease in F-measure. In most of the datasets, the top-performing method wins by a slight margin. However, in some cases, a method may have the highest F-measure but not the highest number of selected features. This exception is observed in only four datasets (breast-w, car, heart-c, and sonar). In these datasets, the method with the highest F-measure also exhibits the best performance for the average number of selected features.

Figure 3 and Figure 4 given in [56] are other ways to represent Table 3. Figure 3 given in [56] depicts the average F-measures of algorithms for 10 runs. BOth figures show the competitive side of the proposed method. Therefore, F-measure values are so close to each other. When dealing with datasets with fewer features, the performance of BASO is similar to that of the other algorithms. However, when dealing with datasets with more features and samples, BASO performs better. As previously mentioned, BASO leverages the information from the past behavior of its members to guide the search. The availability of more features and samples increases the opportunities for accurate and appropriate inference when evaluating the past behavior of the members. In Figure 4, given in [56] it can be observed that the number of selected features is also close among the methods, except for BGWO, which generally selects a higher number of features as

Table 3. Average F-measure and average number of selected features

Datasets	F-measure without FS	Average F-measure				Average No. of selected features			
		BASO	BGA	BPSO	BGWO	BASO	BGA	BPSO	BGWO
Breast-w	0.969	0.982	**0.983**	0.981	0.981	6.5	6.3	6.5	7.1
Car	0.925	0.920	**0.922**	0.919	0.911	5	5	5	5.2
Credit-g	0.721	0.829	**0.831**	0.824	0.825	11.1	10.6	9.9	13.9
Dermato.	0.959	0.988	0.988	0.987	**0.989**	19.7	19.4	20	25.6
Glass	0.708	**0.778**	0.750	0.753	0.754	7.6	6.3	7.8	7.4
Heart-c	0.812	**0.857**	0.854	0.855	0.834	4.8	5.4	5	6.8
Ionosph.	0.859	**0.887**	**0.887**	0.870	0.853	11.4	11.4	11.1	11.7
Lymph.	0.797	**0.896**	0.891	0.893	0.868	10.8	10.2	9.9	13.3
Sonar	0.860	**0.892**	0.889	0.880	0.865	27.5	28.8	28.9	41.6

a feature subset. Although this occasionally puts BGWO at the top of the performance list, it is worth sacrificing a small amount of performance to deal with fewer features.

Having both close best and worst cases, as can be observed in Table 4 given in [56], is another aspect that supports the competitiveness of the BASO method. The table shows the number of features used to achieve the related performance, indicated in parentheses. While the BGA method has achieved most of the best cases according to the table, this does not necessarily reflect its overall performance. Due to the inherent randomness of the algorithms, some methods may occasionally attain the best score by chance. To provide a comprehensive assessment and avoid any reliance on chance, the methods are run multiple times and the average score of the runs is considered. For almost all of the datasets, BGWO selects the largest number of features. The best of the best cases and the best of the worst cases are shown in bold.

To compare the performance of traditional FS methods with that of BASO, we have created Table 5 given in [56]. As shown in the table, except for the car and ionosphere datasets, BASO outperforms CHI, IG, GR, and ReliefF in terms of performance. It should be noted that when the number of features selected by BASO is not an integer, it is rounded up for traditional FS methods, which may provide some advantages to the traditional methods.

Figure 5 given in [56] shows the F-measure obtained from the dataset with all features and selected features. As shown in the figure, the proposed method increases the F-measure value while decreasing the number of features for all datasets except the car dataset. The convergence of the proposed method was also tested, and presented in Figure 6 given in [56]. In the figure, it can be seen how fast the method converges in terms of the number of iterations. In this figure, it can also be seen that it is appropriate not to choose the iteration number too high since there is no increase in the F-measure values after about 20 iterations.

6. Conclusions

In this study, the Anarchic Society Optimization (ASO) algorithm, which simulates anarchic communities, is proposed as a binary approach for feature selection (FS). This is the first study that uses the ASO algorithm as binary for the FS domain. The fact that the proposed algorithm is based on human behavior and has not been used in the FS problem before, motivated authors to carry out this study. The performance of the proposed approach is evaluated using nine benchmark datasets from the UCI machine learning repository. To compare performance, binary versions of three common optimization algorithms, namely Binary Genetic Algorithm (BGA), Binary Particle Swarm Optimization (BPSO), and Binary Grey Wolf Optimizer (BGWO), are employed and applied to the same datasets. The aspect that distinguishes BASO from other algorithms is its attention to the past, alongside personal best and global best. The F-measure results are so close to each other according to the experiments. This shows the ambitious aspect of the proposed BASO. Furthermore, the results indicate that BASO can infer more meaningful correlations for the past from medium or large datasets, as compared to small datasets. Among the nine benchmark datasets, BASO outperforms the other algorithms for five of them. BASO is the second position for the remaining four datasets, with a difference of 0.001-0.002 in terms of average F-measure, as compared to the top algorithm.

The experiments demonstrate that ASO's supervised randomness facilitates fast convergence. Traditional FS methods, such as Chi-Square, Information Gain, Gain Ratio, and ReliefF, were applied to the same datasets. BASO outperforms these methods in terms of F-measure and the number of selected features. This study suggests that deviating from the pursuit of the best solution and allowing the swarm to be anarchical, adventurous, and fickle can yield good results. BASO's these three characteristics enable it to overcome local optima and explore uncharted solution spaces, even when the swarm situation worsens.

References

[1] M.-C. CRISTESCU, *Machine learning techniques for improving the performance metrics of functional verification*, Romanian Journal of Information Science and Technology **24**(1), 2021, pp. 99–116.

[2] I.-D. BORLEA, R.-E. PRECUP and A-B. BORLEA, *Improvement of K-means cluster quality by post processing resulted clusters*, Procedia Computer Science **199**, 2022, pp. 63–70.

[3] C. POZNA and R.-E. PRECUP, *Aspects concerning the observation process modelling in the framework of cognition processes*, Acta Polytechnica Hungarica **9**(1), 2012, pp. 203–223.

[4] R.-E. PRECUP, C.-A. BOJAN-DRAGOS, E.-L. HEDREA, R.-C. ROMAN and E.-M. PETROU, *Evolving fuzzy models of shape memory alloy wire actuators*, Romanian Journal of Information Science and Technology **24**(4), 2021, pp. 353–365.

[5] I. A. ZAMFIRACHE, R.-E. PRECUP, R.-C. ROMAN and E. M. PETRIU, *Policy iteration reinforcement learning-based control using a grey wolf optimizer algorithm*, Information Sciences **585**, 2022, pp. 162–175.

[6] S. SOGUTCU, M. INAL, C. CELIKHASI, U. YILDIZ, N. O. DOGAN and M. PEKDEMİR, *Early detection of mortality in COVID-19 patients through laboratory findings with factor analysis and artificial neural networks*, Romanian Journal of Information Science and Technology **25**(3-4), 2022, pp. 290–302.

[7] J. HAN, J. PEI and M. KAMBER, *Data Mining: Concepts and Techniques*, Elsevier, 2011.

[8] S. PALANISAMY and S. KANMANI, *Artificial bee colony approach for optimizing feature selection*, International Journal of Computer Science Issues **9**(3), 2012, paper 432.

[9] E. ZORARPACI and S. A. OZEL, *A hybrid approach of differential evolution and artificial bee colony for feature selection*, Expert Systems with Applications **62**, 2016, pp. 91–103.

[10] E. TALBI, *Metaheuristics: From Design to Implementation*, John Wiley & Sons, 2009.

[11] D. KARABOGA and B. BASTURK, *A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm*, Journal of Global Optimization **39**(3), 2007, pp. 459–471.

[12] M. DORIGO and G. DI CARO, *Ant colony optimization: a new meta-heuristic*, Proceedings of 1999 Congress on Evolutionary Computation, Washington, DC, USA, 1999, pp. 1470–1477.

[13] S. MIRJALILI, *The ant lion optimizer*, Advances in Engineering Software **83**, 2015, pp. 80–98.

[14] M. NAGHDIANI and M. JAHANSHAHI, *GSO: A new solution for solving unconstrained optimization tasks using Garter Snake's behavior*, Proceedings of 2017 International Conference on Computational Science and Computational Intelligence, Las Vegas, NV, USA, 2017, pp. 328–333.

[15] C. ZHAOHUI and T. HAIYAN, *Notice of retraction: cockroach swarm optimization*, Proceedings of 2nd International Conference on Computer Engineering and Technology, Chengdu, China **6**, 2010, pp. 652–655.

[16] J. KENNEDY and R. EBERHART, *Particle swarm optimization*, Proceedings of ICNN'95-International Conference on Neural Networks, Perth, WA, Australia **4**, 1995, pp. 1942–1948.

[17] X-S. YANG and S. DEB, *Eagle strategy using Lévy walk and firefly algorithms for stochastic optimization*, Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Springer, 2010, pp. 101–111.

[18] J. LAMY, *Artificial Feeding Birds (AFB): a new metaheuristic inspired by the behavior of pigeons*, in S. Shandilya, S. Shandilya, and A. Nagar (Eds.), Advances in Nature-Inspired Computing and Applications, Springer, Cham, 2019, pp. 46–61.

[19] , S. MIRJALILI, S. M. MIRJALILI and A. LEWIS, *Grey wolf optimizer*, Advances in Engineering Software **69**, 2014, pp. 46–61.

[20] S. DEB, S. FONG and Z. TIAN, *Elephant search algorithm for optimization problems*, Proceedings of 2015 Tenth International Conference on Digital Information Management, Jeju, South Korea, 2015, pp.249–255.

[21] G. DHIMAN and V. KUMAR, *Spotted hyena optimizer for solving complex and non-linear constrained engineering problems*, in N. Yadav, A. Yadav, J. Bansal, K. Deep and J. Kim (Eds.), Harmony Search and Nature Inspired Optimization Algorithms, Springer, Singapore, 2019, pp. 857–867.

[22] V. HAYYOLALAM and A. A. P. KAZEM, *Black widow optimization algorithm: A novel meta-heuristic approach for solving engineering optimization problems*, Engineering Applications of Artificial Intelligence **87**, 2020, aper 103249.

[23] S. MIRJALILI and A. LEWIS, *The whale optimization algorithm*, Advances in Engineering Software **95**, 2016, pp. 51–67.

[24] S. SHADRAVAN, H. R. NAJI and V. K. BARDSIRI, *The sailfish optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems*, Engineering Applications of Artificial Intelligence **80**, 2019, pp. 20–34.

[25] S.A. UYMAZ, G. TEZEL and E. YEL, *Artificial Algae Algorithm (AAA) for nonlinear global optimization*, Applied Soft Computing **31**, 2015, pp. 153–171.

[26] F. MERRIKH-BAYAT, *The runner-root algorithm: A metaheuristic for solving unimodal and multimodal optimization problems inspired by runners and roots of plants in nature*, Applied Soft Computing **33**, 2015, pp. 292–303.

[27] A. AHMEDI-JAVID, *Anarchic society optimization: A human-inspired method*, Proceedings of 2011 IEEE Congress of Evolutionary Computation, New Orleans, LA, USA, 2011, pp. 2586–2595.

[28] J. ZHANG, M. XIAO, L. GAO and Q. PAN, *Queuing search algorithm: A novel metaheuristic algorithm for solving engineering optimization problems*, Applied Mathematical Modelling **63**, 2018, pp. 464–490.

[29] E. FATTAHI, M. BIDAR and H. R. KANAN, *Focus group: an optimization algorithm inspired by human behavior*, International Journal of Computational Intelligence and Applications **17**(1), 2018, pp. 1850002.

[30] J. TOO, A. R. ABDULLAH, M. SAAD, N. M. ALI and W. TEE, *A new competitive binary grey wolf optimizer to solve the feature selection problem in EMG signals classification*, Computers **7**(4), 2018, 58.

[31] J. TOO, A. R. ABDULLAH and M. SAAD, *A new co-evolution binary particle swarm optimization with multiple inertia weight strategy for feature selection*, Informatics **6**(2), 2019, paper 21.

[32] J. TOO and A. R. ABDULLAH, *A new and fast rival genetic algorithm for feature selection*, The Journal of Supercomputing **77**, 2020, pp. 2844–2874.

[33] J. H. HOLLAND, *Adaptation in Ntural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, MIT Press, 1992.

[34] W. SHANG, H. HUANG, H. ZHU, Y. LIN, Y. QU and Z. WANG, *A novel feature selection algorithm for text categorization*, Expert Systems with Applications **33**(1), 2007, pp. 1–5.

[35] M. TARADEH, M. MAFARJA, A. A. HEIDARI, H. FARIS and I. ALJARAH, S. MIRJALILI and H. FUJITA, *An evolutionary gravitational search-based feature selection*, Information Sciences **497**, 2019, pp. 219-239.

[36] M. KAYA KELES and U. KILIC, *Artificial bee colony algorithm for feature selection on SCADI dataset*, Proceedings of 3rd International Conference on Computer Science and Engineering), Sarajevo, Bosnia and Herzegovina, 2018, pp. 463–466.

[37] U. KILIC and M. KAYA KELES, *Feature selection with artificial bee colony algorithm on Z-Alizadeh sani dataset*, Proceedings of 2018 Innovations in Intelligent Systems and Applications Conference, Adana, Turkey, 2018, pp. 1–3.

[38] M. KAYA KELES and U. KILIC, *Classification of brain volumetric data to determine Alzheimer's disease using artificial bee colony algorithm as feature selector*, IEEE Access **10**, 2022, pp. 82989-83001.

[39] M. KAYA KELES, A. E. KELES and U. KILIC, *Prediction of concrete strength with data mining methods using artificial bee colony as feature selector*, Proceedings of 2018 International Conference on Artificial Intelligence and Data Processing, Malatya, Turkey, 2018, pp. 1–4.

[40] M. KAYA KELES, A. E. KELES and U. KILIC, *Proposed artificial bee colony algorithm as feature selector to predict the leadership perception of site managers*, The Computer Journal **64**(3), 2020, pp.408–417.

[41] M. M. MAFARJA, AND S. MIRJALILI, *Hybrid whale optimization algorithm with simulated annealing for feature selection*, Neurocomputing **260**, 2017, pp. 302–312.

[42] S. ARORA and P. ANAND, *Binary butterfly optimization approaches for feature selection*, Expert Systems with Applications **116**, 2019, pp. 147–160.

[43] M. KAYA KELES and U. KILIC, *Binary black widow optimization approach for feature selection*, IEEE Access **10**, 2022, pp. 95936–95948.

[44] M. MAFARJA, A. QASEM, A. A. HEIDARI, I. ALJARAH, H. FARIS and S. MIRJALILI, *Efficient hybrid nature-inspired binary optimizers for feature selection*, Cognitive Computation **12**(1), 2020, pp. 150–175.

[45] M. QARAAD, S. AMJAD, N. K. HUSSEIN and M. A. ELHOSSEINI, *Large scale salp-based grey wolf optimization for feature selection and global optimization*, Neural Computing and Applications **34**, 2022, pp. 8989–9014.

[46] P. HU, J. PAN, S. CHU and C. SUN, *Multi-surrogate assisted binary particle swarm optimization algorithm and its application for feature selection*, Applied Soft Computing **121**, 2022, pp. 108–736.

[47] A. AHMADI-JAVID and P. HOOSHANGI-TABRIZI, *An anarchic society optimization algorithm for a flow-shop scheduling problem with multiple transporters between successive machines*, Proceedings of 2012 International Conference on Industrial Engineering and Operations Management, Istanbul, Turkey, 2012, pp. 1913–1921.

[48] A. AHMADI-JAVID and P. HOOSHANGI-TABRIZI, *A mathematical formulation and anarchic society optimisation algorithms for integrated scheduling of processing and transportation operations in a flow-shop environment*, International Journal of Production Research **53**(19), 2015, pp. 5988–6006.

[49] A. AHMADI-JAVID and P. HOOSHANGI-TABRIZI, *Integrating employee timetabling with scheduling of machines and transporters in a job-shop environment: A mathematical formulation and an anarchic society optimization algorithm*, Computers & Operations Research **84**, 2017, pp. 73–91.

[50] H. SHAYEGHI and J. DADASHPOUR, *Anarchic society optimization based PID control of an Automatic Voltage Regulator (AVR) system*, Electrical and Electronic Engineering **2**, 2012, pp. 199–207.

[51] A. BOZORGI, O. BOZORG-HADDAD, R. MOHAMMAD-MEHDI, M. LATIFI and X. CHU, *Applications of the Anarchic Society Optimization (ASO) algorithm for optimizing operations of single and continuous multi-reservoir systems*, Journal of Water Supply: Research and Technology—AQUA **66**, 2017, pp. 556–573.

[52] O. BOZORG-HADDAD, M. LATIFI, A. BOZORGI, M. RAJABI, S. T. O. NAEENI, and H. LOAICIGA, *Development and application of the Anarchic Society Algorithm (ASO) to the optimal operation of water distribution networks*, Water Science and Technology: Water Supply **18**(1), 2017, pp. 318–332.

[53] S. S. SHREEM, H. TURABIEH and F. BAOTHMAN, *Enhanced binary genetic algorithm as a feature selection to predict student performance*, Soft Computing **26**, 2022, pp. 1811–1823.

[54] F. SOUZA, C. PREMEBIDA and R. ARAUJO, *High-order conditional mutual information maximization for dealing with high-order dependencies in feature selection*, Pattern Recognition **131**, 2022, paper 108895.

[55] B. COOIL, R. S. WINER and D. L. RADOS, *Cross-validation for prediction*, Journal of Marketing Research **24**(3), 1987, pp. 271–279.

[56] U. KILIC, E. SARAC ESSIZ and M. KAYA KELES, *Supplementary material of the paper Umit Kilic, Esra Sarac Essiz, Mumine Kaya Keles, "Binary Anarchic Society Optimization for Feature Selection", Romanian Journal of Information Science and Technology, 2023*. Accessed: July 18, 2023. [Online]. Available: https://github.com/ukilic-edu/ROMJIST/blob/main/Supplementary-material-ROMJIST.pdf.