A Binary Grasshopper Optimization Algorithm for Feature Selection

Reyhaneh Yaghobzadeh[1], Seyyed Reza Kamel *[1], Mojtaba Asgari[1], Hassan Saadatmand[2]
[1.] Department of Computer Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran.
[2.] Department of Electric, Mashhad Branch, Mashhad University, Mashhad, Iran.

Abstract:

Introduction: The aim of the feature selection is to find the most important information from a set of given features. It has been used widely in recent years. The set of features primarily has associated features that influence classification performance and increase training time. So, the feature selection is necessary to remove inappropriate features and improve classification performance. In this paper, a nature-inspired feature selection method is proposed based on the behavior of grasshoppers. Grasshopper optimization algorithm is expressed for continuous problems as standard.

Methods: In this paper, continuous grasshopper optimization algorithm is converted into binary by using a nonlinear mapping function. Also, the binary grasshopper optimization algorithm will be converted into a powerful algorithm by combining and retaining exploration and exploitation.

Result: the dataset taken from the UCI repository is used. By comparing the proposed method with GA, Binary Bat Algorithm (BBA), we indicate that it has had better performance than previous methods in terms of accuracy, reduction rate, precision, and computational time.

Conclusion: In this paper, the feature selection problem is presented by the binary grasshopper optimization algorithm. The results are compared at different algorithms. It was concluded that the binary grasshopper optimization algorithm is the best method for feature selection

Keywords: Feature Selection, Binary Grasshopper Optimization Algorithm, Classification.

1. INTRODUCTION

The main purpose of feature selection is to understand the basic process that creates the data to select a subset of the relevant features of the available items[1]. High-dimensional data offer serious challenges for existing learning practices [2]. To reduce the dimension, feature selection uses a wide technique. It selects a small subset of the relevant features from the real set, which usually improves the educational performance of the model. Using the feature selection technique, the cost of learning computing could also be reduced [2]. To date, many feature selection techniques have used meta heuristic algorithms to search for optimal subset such as BBA [3], binary dragonfly algorithm [4], and so on. Meta heuristic algorithms have two concepts of exploration and exploitation, by preserving and combining these two concepts, the algorithm can be converted into a powerful one. Unfortunately, in other algorithms there is no proper interaction between these two concepts. For example, the genetic algorithm combines these two concepts but failed to establish a proper interaction between the two. Given the features such as easy implementation and quick synchronization, grasshopper optimization algorithm has attracted more attention. In this paper, the feature selection is discussed by combining these two concepts and using the binary grasshopper optimization algorithm. The mentioned algorithm is also able to solve real problems with unknown space [19]. In the second section, we will discuss the literature review, in the third section we will present the proposed method and in the fourth section we will analyze and evaluate the results.

2. LITERATURE REVIEW

In this section, we will review the literature on the metaheuristic and binary algorithms for feature selection.

In [3], a binary version of the bat algorithm is proposed. A comparative study with two dimensions of PSO and GA with more than 22 benchmark function has been conducted to achieve the result. In addition, the Wilcoxon non-parametric statistical test was significant at a significant level of 5% to determine whether the results of the proposed algorithm differ from other algorithms. The results suggest that the proposed binary algorithm BBA can significantly outperform others in most benchmark functions.

In [5], a version of the PSO algorithm is presented that operates on discrete binary variables. In the binary version, trajectories are variations in the probability of coordination occurrence in a value of zero or one. The proposed algorithm is able to solve many different problems very quickly. The new version of the algorithm enhances the capabilities of the continuous-mode algorithm and can improve each continuous or discrete function.

In [6], a nature-inspired feature selection technique has been proposed based on the behavior of bats. This view combines exploration power with OPF classification rate to find a set of features that maximize precision in a valid set. The algorithm is implemented on 5 datasets and compared with the PSO, FFA and GSA algorithms. It has been more successful in three datasets than other algorithms, and in the other two, this algorithm has been the second successful algorithm.

In [7], a new feature selection method is proposed based on cuckoo's behavior, which is the binary version of the cuckoo search algorithm. This algorithm has been implemented on 2 datasets based on the concept of the thief exploration in power distribution systems and its superiority over other nature-inspired optimization techniques has been proven. This algorithm is also perfect for feature selection and quickest approach to industrial datasets.

In [7], a new binary version of the synthetic Binary Coordinate Ascent algorithm based on genetic operators (GA-ABC) is proposed to solve binary optimization problems. In this method, the global-localized search capability of the primary Artificial Bee Colony Algorithm (ABC) in a binary domain has been improved. The efficiency of this algorithm has been tested on two well-known optimization problems: dynamic image clustering and 0-1 Knapsack problems. The results show that GB-ABC algorithm is the best algorithm for binary optimization compared to other binary optimization algorithms.

In [8], a wrapper feature selection algorithm is proposed based on the binary dragonfly algorithm. Dragonfly algorithm is a new particle swarm algorithm that emulates the behavior of dragonflies. The results of this algorithm are compared with PSO and GA algorithms and prove that this algorithm has a high performance than the other two ones.

In [9], both the binary genetic and the Floating Point (FP) algorithms are examined using a dynamic control problem. The results show that the FP presentation of the genetic algorithm is faster and more consistent than its binary mode and is more accurate than it. Also, FP mode can achieve a higher performance than binary mode by certain operators.

In [10], a new feature selection algorithm is proposed based on Ant Colony Optimization Algorithm (ACO). In this algorithm, the efficiency of the classifier and the selected vector length are taken as the ACO heuristic information. This algorithm selects the subset of the optimal feature, without using the previous information of the features, and its implementation is easy and its computational complexity is very low due to the use of a simple classifier. The results show that this algorithm has a better performance than genetic-based algorithms with fewer features and low implementation time.

In [11], a new binary version of the Gray Wolf Optimization Algorithm (GWO) is presented which selects the subset of the optimal feature subset for the purposes of classification. This algorithm has been compared with PSO and GA algorithms and its ability to search for feature space has been proven to find optimal features. Also, the efficiency of the features selected by this algorithm is higher than that of two other algorithms.

In [12], binary optimizer types of Ant Lion Optimizer (ALO) algorithm are proposed, which selects a subset of optimal features for the purpose of the classification. This algorithm imitates the process of ant lions hunting. This algorithm can search optimally for the feature space and achieve a better solution than PSO, GA, ALO, and BBA.

In [13], a new spam detection method is proposed that focuses on reducing the incorrect labeling error of non-spam tags as spam. In this technique, the wrapper feature selection method is used. The results indicate that this method is effective and reduces the incorrect labeling error.

In [14], the rate of the harmony search algorithm and Optimal Path Forest (OPT) classifier has been used to arrive at a quick and accurate new perspective for feature selection. Compared to other pattern recognition and feature selection algorithms, this proposed hybrid algorithm performs better.

In [15], a new hybrid approach is proposed for simultaneous feature selection and weighting features based on the K-NN law on the TS heuristic algorithm. The results show a great improvement in the efficiency and classification accuracy. Experiments also show that the TS heuristic algorithm is more powerful than the simple TS algorithm and sequential search algorithms.

In [16], a new perspective is proposed based on the Gravitation Search Algorithm (GSA) for feature selection. This algorithm combines the GSA optimization behavior with the rate of the OPF classifier to provide a precise and fast framework for features selection. The results indicate that this algorithm is as powerful as other similar algorithms.

In this section, the feature selection meta heuristic algorithms were investigated, and it was shown that each of these algorithms has some disadvantages for feature selection. In the next section, we will present the proposed method.

3. METHODS

In the past, the research literature and their problems were reviewed. This section provides a brief explanation of grasshopper optimization algorithm and the proposed method.

3.1. Standard grasshopper optimization algorithm

Grasshoppers are from the insect family. They are known as pests that cause damage to agricultural products. Grasshopper life cycle is shown in Figure (1)[17].

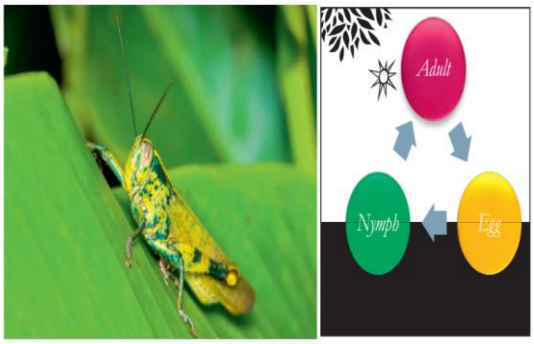

Fig. 1. Grasshopper life cycle

Equation (1) has been used to model the behavior of grasshoppers [18].

$$X_i = S_i + G_i + A_i \tag{1}$$

Where X_i is the location of grasshoppers, S_i is the social interaction between the grasshoppers, G is the gravitational force of the grasshoppers and A defines the direction of the wind direction. These show the location of a grasshopper. To create random behavior, we can use the equation (2), in which r can randomly vary between 0 and 1:

$$X_i = r_1 S_i + r_2 G_i + r_3 A_i \tag{2}$$

Si is the social interaction that is calculated according to equation (3), where d_{ij} donates the distance between *i-th* grasshopper and *j-th* grasshopper.

$$\mathrm{S_i} = \sum_{\substack{j=1 \\ j \neq i}}^{N} \mathrm{s}\,(\mathrm{d_{ij}})\widehat{\mathrm{d_{ij}}} \tag{3}$$

In equation (3), d_{ij} denotes the distance between *i-th* grasshopper and *j-th* grasshopper, and is calculated as $d_{ij} = |X_j - X_i|$. S function is a mapping for the distance between the grasshoppers obtained according to equation (4):

$$\mathrm{s}\,(\mathrm{r}) = \mathrm{fe}^{\frac{-\mathrm{r}}{\mathrm{l}}} - \mathrm{e}^{-\mathrm{r}} \tag{4}$$

In which f represents the intensity of gravity, the general formula of which is in accordance with equation (5):

$$\mathrm{X_i} = \sum_{\substack{j=1 \\ j \neq i}}^{N} \mathrm{s}\,(|\mathrm{x_j} - \mathrm{x_i}|)\frac{\mathrm{x_j} - \mathrm{x_i}}{\mathrm{d_{ij}}} - \mathrm{g}\widehat{\mathrm{e_g}} + \mathrm{u}\widehat{\mathrm{e_w}} \tag{5}$$

In this equation, N is the number of grasshoppers, but since grasshoppers move on the ground, their location should not be more than a threshold, and thus the modified equation (6) is used:

$$X_i^d = c\left(\sum_{\substack{j=1 \\ j \neq i}}^{N} c\frac{ub_d - lb_d}{2} s\,(|x_j^d - x_i^d|)\frac{x_j - x_i}{d_{ij}}\right) + \widehat{T_d} \tag{6}$$

This equation of the location of a grasshopper is defined based on its current location, target location, and location of all other grasshoppers. In this equation, C is one of the important parameters of the grasshopper optimization algorithm, a reduction factor, by which the safe, and repulsion and gravity regions are affected. The update for this parameter is obtained from equation (7).

$$c \ = cmax - l\frac{cmax - cmin}{L} \tag{7}$$

The steps of the standard grasshopper optimization algorithm are shown in Figure (2).

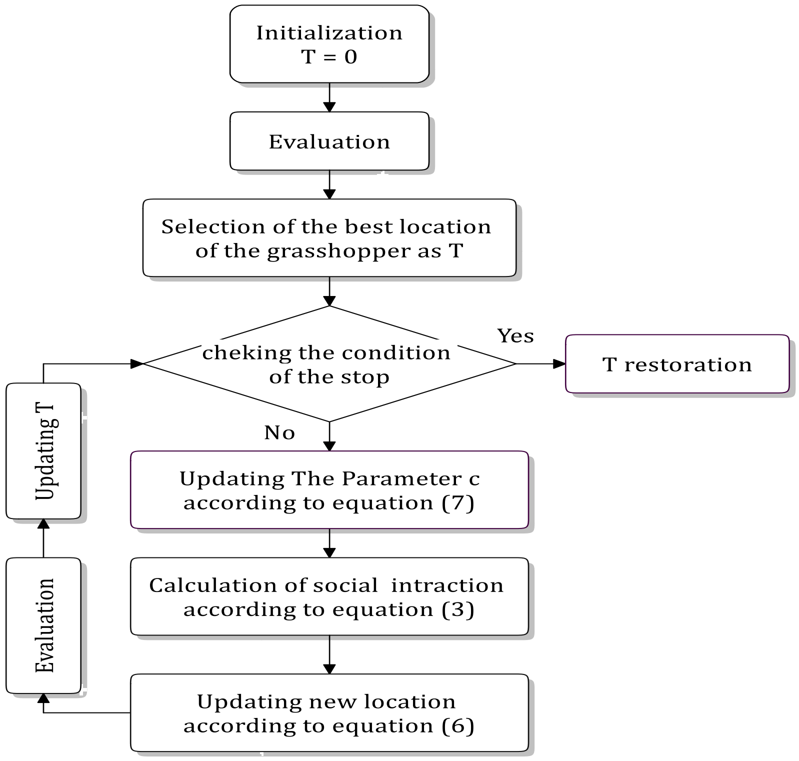
Initialization
T = 0
Evaluation
Selection of the best location of the grasshopper as T
cheking the condition of the stop
Yes
T restoration
No
Updating The Parameter c according to equation (7)
Calculation of social intraction according to equation (3)
Updating new location according to equation (6)
Evaluation
Updating T

Fig. 2. Grasshopper optimization algorithm function

3.2. Binary grasshopper optimization algorithm

In the standard grasshopper optimization algorithm, solutions are updated to continuous values. For feature selection in this algorithm, the search space is modeled as a Boolean network. A solution to the problem of selecting or not selecting a feature is using the binary vector. This algorithm consists of a vector containing zero and one, in which, one indicates that the feature is selected and zero indicates that the feature is not selected. We construct this binary vector according to Figure 3. As shown in Figure 3, this algorithm is transformed into a binary grasshopper optimization algorithm in three main steps :

1. update the parameter c,
2. nonlinear mapping of v,
3. updating new location of grasshoppers.

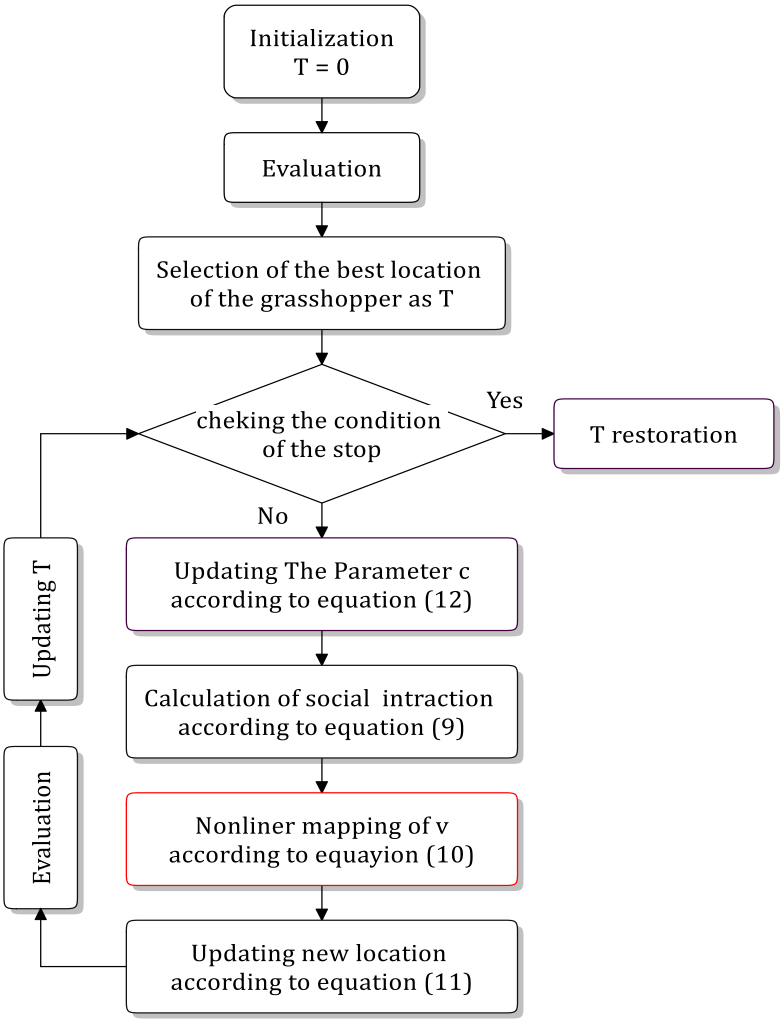
Initialization
T = 0
Evaluation
Selection of the best location of the grasshopper as T
cheking the condition of the stop
Yes
T restoration
No
Updating The Parameter c according to equation (12)
Calculation of social intraction according to equation (9)
Nonliner mapping of v according to equayion (10)
Updating new location according to equation (11)
Updating T
Evaluation

Fig. 3. Binary grasshopper optimization algorithm steps

To construct this algorithm, the modified equations (8), (9) and (10) are used. In Equation (8), instead of using $G + A$, a T label is used. T is the best grasshopper that has ever existed. Equation (9) shows the social interaction of grasshoppers' behavior in which:

$$X_i^d = c\, s_i + T_d \tag{8}$$

$$\mathrm{S_1} = \sum_{\mathrm{i\#j}}^{\mathrm{N}} \frac{\mathrm{c}}{2} \cdot \left|\mathrm{X_i - X_j}\right| \mathrm{d_j} \tag{9}$$

Since the decision variable in binary grasshopper optimization algorithm is only between 0 and 1, we use a nonlinear function called V that maps the social behavior of grasshoppers to another function or space and is calculated in accordance with equation (10).

$$V_i = \left|\frac{2}{\pi} \tan^{-1}\left(\frac{10\pi}{2}\, c\, S_i\right)\right| \tag{10}$$

Finally, we need to update the location of grasshoppers. Equation (6) is obtained from this change and modification and is calculated according to Equation (11).

$$X_i = \begin{cases} T_i & p > v_i \\ \overline{X_\imath} & p < v_i \end{cases} \tag{11}$$

The proposed mathematical formulation is able to perform exploration and exploitation in search space. However, there should be a mechanism to adjust the level of exploration and exploitation of search factors.

The reason for the simultaneous use of exploration and exploitation in this algorithm is to have an overview of the search space initially and to consider all the possible solutions and a partial look at the desired space at the end of the algorithm to find the optimal solution. In nature, locusts move first and seek food locally, because they do not have wings at the larva stage. Then they move freely in the air and explore a region with a larger scale. In the case of using an exploration algorithm without exploitation, only a random search is performed that is not appropriate for final implementations and obtaining the final optimal solution. Also, the use of an exploitation algorithm without considering an exploratory algorithm leads to finding a part of the optimal local solutions. However, in randomized optimization algorithms, optimal exploitation is done first due to the need to find more promising areas of search space. After finding more promising areas, exploitation requires search factors to search locally to find a good approximation of the global optimum. In equation (11), T contributes to exploitation and convergence, and X helps preserving the diversity and exploration.

3.3. Binary grasshopper optimization algorithm with the parameter c update

Updating the parameter C plays an important role in the interaction between the two concepts of exploration and exploitation. Updating this parameter is obtained in accordance with equation (12):

$$c = cinf + \left((1 - it^{w})^{(1/w)}\right) * (1 - cinf) \qquad (12)$$

In the above equation, cinf is a value close to zero (approximately 0.0004), which shows the final value of c that ends it. It also has the number of algorithm repetition and the values of w vary between 0 and 1. The switch between exploration and exploitation in this algorithm is performed by changing the value of the parameter w between 0 to 1. That is, if w is greater than 0.5, exploration is more important, but if w is less than 0.5, exploitation is more important. How to update the parameter c is shown in Figure 4.

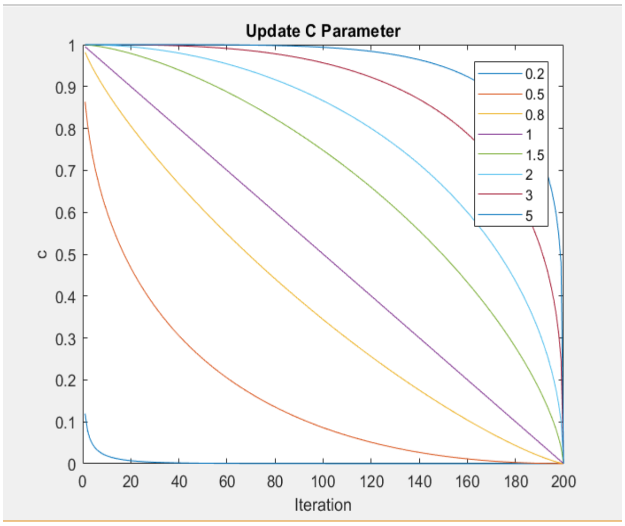

Fig. 4. Updating the parameter c

As shown in Figure 4, changing the parameter value gives different charts. For example, with a value of 0.5 immediately with 60 iteration it has the value of 0.2, and actually ignores the exploitation. If it is equal to 1, then an interaction is established linearly between the two concepts. If we want to approach it to the concept of exploitation, we need to set the value at more than one, for example, equal to 2, which is here 0.9 at 60 iteration.

In the next section, we analyze and evaluate the datasets used, the settings and benchmarks for comparing, analyzing and evaluating the overall results.

4. RESULT

In this section, we analyze the data used, as well as analyze the results and benchmarks evaluated.

4.1. Datasets

In this study, we use 7 datasets according to Table 1.

Table 1. Dataset

Number of features	Number of records	Datasets	Row
9	788	diabet	1
10	699	bcw	2
41	1055	biodeg	3
19	155	hepatitis	4
41	208	sonar	5
57	4061	spambase	6
56	303	Z_Alizadeh_sani	7

4.2. Settings and comparison benchmarks

To perform simulation operations, the datasets are first entered the simulation environment and classified after normalization and validation. In the end, the results and benchmarks are compared with all the classified data. In order to analyze the performance of the proposed system, four factors of accuracy, precision, computational time and reduction rate are used as follows:

- **Accuracy:** The most important benchmark to determine the performance of a classification algorithm. It shows how many percent of the total set of test records is properly categorized. The accuracy benchmark is calculated according to equation (13).

$$Accuracy = \frac{TP + TN}{TP + \mathrm{TN} + FP + FN} \tag{13}$$

- **Precision**: Indicates the algorithm ability to recognize the positive category. Here, the positive category is the patient one.

$$perecision = \frac{TP}{TP + FN} \tag{14}$$

- **Computational time**: indicates the time required to run the application. The less computational time the algorithm performs better.
- **Reduction rate**: shows the reduction percentage in the features and is obtained from equation (15).

$$\mathrm{Reduction\ rate} = 1 - \frac{\mathrm{ns}}{\mathrm{nf}} \tag{15}$$

According to equation (15), ns is the number of features selected by algorithm and nf is the total number of features.

5. DISCUSSION

In this section, we analyze and evaluate the results obtained from the four methods of feature selection with three different classifiers on different datasets.

5.1. Accuracy benchmark

In this section, we examine the accuracy benchmark in the different datasets in four methods of feature selection. This comparison is shown in Table 2. As seen in the table in the diabetes dataset [18], the use of the feature selection technique of BBA along with the Tree classifier provides better accuracy than other algorithms. Among the advantages of this algorithm in features selection on a diabetes dataset is having a high convergence rate compared to other meta heuristic algorithms. It also uses frequency regulation to solve problems and provides the ability to work with high-dimensional data.

Table 2. Comparison of the accuracy benchmark

The datasets used Accuracy							**Feature Selection Algorithms**	**Row**
Alizadeh	**Spambase**	**Sonar**	**Hepatit**	**Biodeg**	**Bcw**	**Diabet**		
73.77	**90.21**	83.33	61.29	**84.83**	97.14	97	**The proposed method**	1
83.50	88.58	**85.71**	61.29	81.04	94.28	95	GA	2
78.68	89.78	71.49	61.29	75.72	94.28	**98.75**	BBA	3
83.60	**89.34**	59.52	**67.74**	75.25	**97.85**	97. 5	**The proposed method with the parameter c update**	4

In the dataset for cancer [19], the BGA with the update of the parameter c has shown a remarkable accuracy, which is due to maintain two benchmarks of exploration and exploitation more than three other algorithms. Hepatitis data are more accurate by using the feature selection method of BGOA_C and the update of the parameter c along with the Tree classifier compared using the other three methods. Here the superior feature of the binomial grasshopper optimization algorithm can be mentioned that is combining and maintaining two benchmarks of exploration and exploitation. In the Sonar dataset [20], the genetic algorithm provides optimal accuracy due to its easy implementation as well as the optimum runtime. In the two spam base and Alizadeh datasets [21], the proposed method offers an optimal accuracy too, by retaining the two concepts of combination and exploitation. In general, it can be concluded that the accuracy benchmark in different datasets has shown a better performance using the proposed method, along with the updating the parameter c.

5.2. The reduction rate and precision benchmarks in the datasets

The reduction rate and precision benchmarks for different datasets and different methods of features selection and the use of Decision TREE classifiers are shown in Table 3. The two reduction rate and precision benchmarks have been placed side by side. In fact, precision is a benchmark that requires calibration, and this calibration is done by the reduction rate, which is the same as the concept of feature selection. In the feature selection, unnecessary data are deleted and the necessary data are identified, so the precision of a model depends on the reduction rate by the feature selection.

Table 3. Comparison of the precision and reduction rate in the feature selection algorithms

The dataset used Percision/Reduction							**Feature Selection Algorithms**	**Row**
Alizadeh	**Spambase**	**Sonar**	**Hepatitis**	**Biodeg**	**Bcw**	**Diabetic**		
P=88.63	P=92.53	P=73.33	P=81.81	**P=78.94**	**P=97.77**	P=96.92	The proposed method	1
R=62%	R=46%	R=45%	R=58%	**R=61%**	**R=60%**	R=54%		
P=84.09	P=89.43	**P=93.33**	P=81.81	P=76.31	P=86.15	P=86.15	GA	2
R=53%	R=62%	**R=64%**	R=79%	R=59%	R=64%	R=64%		
P=84.09	P=91.83	P=60	P=81.81	P=60.52	**P=97.77**	**P=98.46**	BBA	3
R=57%	R=53%	R=60%	R=79%	R=49%	**R=60%**	**R=44**		
P=84.09	**P=93.80**	P=46.66	**P=90.90**	P=77.63	P=98.89	P=95.38	The proposed method with the parameter c update	4
R=42%	**R=50%**	R=60%	**R=70%**	R=49%	R=20%	R=54%		

In the same condition, the algorithm shows a higher performance if the precision rate is higher and also the reduction rate is lower. As shown in Table 3, the BBA performs better than the other ones for the diabetes dataset, which contains 788 records and 9 features. The reason for this can be attributed to the advantage of the BBA, which uses parameter control compared to most of the meta heuristic algorithms that use constant parameters. Also, this algorithm performs better in the dataset that contains large amounts of noise, such as the diabetes dataset. In the cancer dataset, the proposed method, the binary grasshopper optimization algorithm, with the parameter c update, with a precision of 98.89 and a significant reduction rate of 20, showed a much better performance than other algorithms. Also in Biodegg [22], Hepatitis [23], Spam base, and Alizadeh datasets, the proposed method, the binary grasshopper optimization algorithm, as well as the BBA with the update of the parameter c, have shown a significant precision and reduction rate. In the sonar dataset, the GA with a precision of 93.33 and a reduction rate of 64 had a better performance. In general, it can be said that the binary grasshopper optimization algorithm has been able to show better performance than other algorithms with updating the parameter c, due to maintaining the two benchmarks of exploration and exploitation with the significant precision and reduction rate.

5.3. The computational time benchmark in the dataset

The time needed to run the algorithms in different data with different methods of feature selection and using the TREE classifier is shown in Table 4.

Table 4. Comparison of the computational time benchmark

The datasets used Accuracy							**Feature Selection Algorithms**	**Row**
Alizadeh	**Spambase**	**Sonar**	**Hepatit**	**Biodeg**	**Bcw**	**Diabet**		
58.71	85.73	59.57	58.014	64.86	31.86	65.05	The proposed method	1
65.53	93.90	63.85	62.42	72.56	33.56	68.92	GA	2
59.28	85.80	58.23	37.46	67.02	38.21	65.90	BBA	3
59.71	85.44	58.10	58.79	66.65	32.24	65.85	The proposed method with the parameter c update	4

The lower the computational time, the algorithm performs better. As can be seen in the table, the binary grasshopper optimization algorithm, with the update of parameter C in different datasets has shown a more efficient and cost-effective performance in terms of runtime than the other algorithms.

5.4. Average accuracy

Figure 5 shows the average accuracy for 4 feature selection methods. As seen in the chart, the average accuracy is GA: 84.2, BBA: 81.4, GoA:83.93, GOA_c 92. As can be seen, grasshopper optimization algorithm with a parameter c update has had the highest average accuracy, and then the GA has had a good average accuracy.

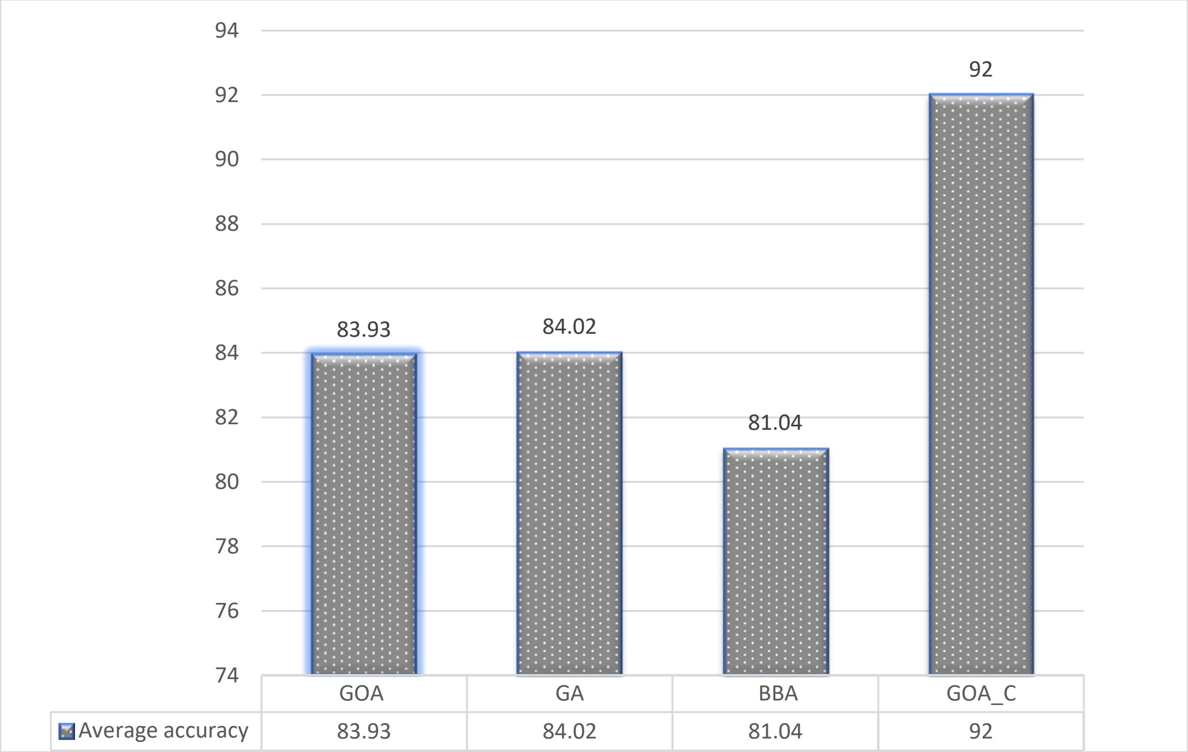

Fig. 5. Average accuracy

6. CONCLUSION

In this paper, the feature selection problem is presented by the binary grasshopper optimization algorithm. The proposed method is performed on 7 datasets taken from the UCI repository and the benchmarks of accuracy, computational time, precision, and reduction rate are investigated. The results are compared at different algorithms. It was concluded that the binary grasshopper optimization algorithm is the best method for feature selection.

REFERENCES

[1] Zawbaa, H.M., E. Emary, and B. Parv. *Feature selection based on antlion optimization algorithm*. in *Complex Systems (WCCS), 2015 Third World Conference on*. 2015. IEEE.

[2] Pandya, R. and J.J.I.J.o.C.A. Pandya, *C5. 0 algorithm to improved decision tree with feature selection and reduced error pruning.* 2015. 117(16): p. 18-21.

[3] Mirjalili, S., et al., *Binary bat algorithm.* 2014. 25(3-4): p. 663-681.

[4] Rodrigues, D., et al. *BCS: A binary cuckoo search algorithm for feature selection*. in *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*. 2013. IEEE.

[5] Kennedy, J. and R.C. Eberhart. *A discrete binary version of the particle swarm algorithm*. in *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on*. 1997. IEEE.

[6] Nakamura, R.Y., et al. *BBA: a binary bat algorithm for feature selection*. in *2012 25th SIBGRAPI conference on graphics, Patterns and Images*. 2012. IEEE.

[7] Ozturk, C., E. Hancer, and D.J.I.S. Karaboga, *A novel binary artificial bee colony algorithm based on genetic operators.* 2015. 297: p. 154-170.

[8] Mafarja, M.M., et al. *Binary dragonfly algorithm for feature selection. in New Trends in Computing Sciences (ICTCS), 2017 International Conference on.* 2017. IEEE.

[9] Mirjalili, S., A.J.S. Lewis, and E. Computation, *S-shaped versus V-shaped transfer functions for binary particle swarm optimization.* 2013. 9: p. 1-14.

[10] Kanan, H.R., K. Faez, and S.M. Taheri. *Feature selection using ant colony optimization (ACO): a new method and comparative study in the application of face recognition system*. in *Industrial Conference on Data Mining*. 2007. Springer.

[11] Emary, E., H.M. Zawbaa, and A.E.J.N. Hassanien, *Binary grey wolf optimization approaches for feature selection.* 2016. 172: p. 371-381.

[12] Emary, E., H.M. Zawbaa, and A.E.J.N. Hassanien, *Binary ant lion approaches for feature selection.* 2016. 213: p. 54-65.

[13] Zhang, Y., et al., *Binary PSO with mutation operator for feature selection using decision tree applied to spam detection.* 2014. 64: p. 22-31.

[14] Ramos, C.C., et al., *A novel algorithm for feature selection using harmony search and its application for non-technical losses detection.* 2011. 37(6): p. 886-894.

[15] Tahir, M.A., A. Bouridane, and F.J.P.R.L. Kurugollu, *Simultaneous feature selection and feature weighting using Hybrid Tabu Search/K-nearest neighbor classifier.* 2007. 28(4): p. 438-446.

[16] Papa, J.P., et al. *Feature selection through gravitational search algorithm*. in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. 2011. IEEE.

[17] Saremi, S., S. Mirjalili, and A.J.A.i.E.S. Lewis, *Grasshopper optimisation algorithm: theory and application.* 2017. 105: p. 30-47.

[18] htpp://archive.ics.uci.edu/ml/machine-learning-databases/pima-indians-diabetes

[19] https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/

[20] https://archive.ics.uci.edu/ml/support/Connectionist+Bench+(Sonar,+Mines+vs.+Rocks)

[21] https://archive.ics.uci.edu/ml/datasets/Z-Alizadeh+Sani

[22] https://archive.ics.uci.edu/ml/machine-learning-databases/00254/