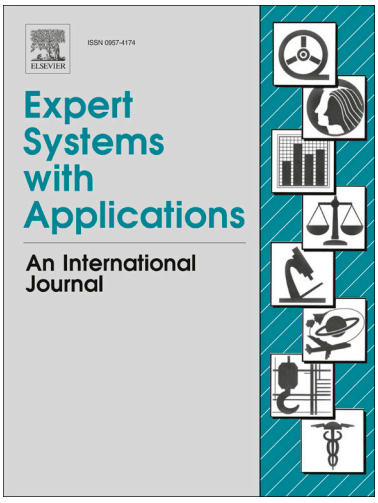
Journal Pre-proofs



An Improved Group Teaching Optimization Algorithm Based on Local Search and Chaotic Map for Feature Selection in High-Dimensional Data

Hamed Khosravi, Babak Amiri, Navid Yazdanjue, Vahide Babaiyan

PII: S0957-4174(22)00822-3
DOI: https://doi.org/10.1016/j.eswa.2022.117493
Reference: ESWA 117493

To appear in: *Expert Systems with Applications*

Received Date: 27 December 2021
Revised Date: 18 April 2022
Accepted Date: 30 April 2022

Please cite this article as: Khosravi, H., Amiri, B., Yazdanjue, N., Babaiyan, V., An Improved Group Teaching Optimization Algorithm Based on Local Search and Chaotic Map for Feature Selection in High-Dimensional Data, *Expert Systems with Applications* (2022), doi: https://doi.org/10.1016/j.eswa.2022.117493

An Improved Group Teaching Optimization Algorithm Based on Local Search and Chaotic Map for Feature Selection in High-Dimensional Data

Hamed Khosravi [1], Babak Amiri [2,*], Navid Yazdanjue [3], Vahide Babaiyan[4]

[1]*School of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran, hamedkhosravi181@gmail.com*

[2]*School of Industrial Engineering, Iran University of Science and Technology, Narmak, Tehran 16846-13114, Iran, babakamiri@iust.ac.ir*

[3]*School of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran, navid.yazdanjue@gmail.com*

[4]*Department of Computer Engineering, Birjand University of Technology, Birjand, Iran, babaiyan@birjandut.ac.ir*

[*]*Corresponding Author: babakamiri@iust.ac.ir*

Abstract

The current study proposes a novel binary group teaching optimization algorithm with local search and chaos mapping (BGTOALC) as a wrapper-based feature selection method to solve high-dimensional feature selection problems. The local search and chaos mapping enhance the performance of the proposed algorithm. Also, two novel binary operators called Binary Teacher Phase Good Group (BTPGG) and Binary Teacher Phase Bad Group (BTPBG) are applied to the teacher's phase for increasing the exploration and exploitation of the algorithm. Moreover, a new Binary Student Opposition-Based Learning (BSOBL) operator is introduced for the student phase, using an opposition-based strategy to achieve better exploitation. Finally, the teacher allocation phase is designed in a binary manner using the new Mean Binary Select (MBS) operator to increase the algorithm's convergence rate. Subsequently, two other binary group teaching optimization algorithms, named BGTOAV and BGTOAS, are developed utilizing the S-shaped and V-shaped transfer functions to compare their performance with the BGTOALC algorithm. The proposed approaches are compared to other state-of-the-art binary algorithms on 30 datasets with different dimensions. Different experiments prove that the BGTOALC method outperforms the previous methods in terms of reducing the number of selected features and increasing the accuracy of the machine learning algorithm. Eventually, statistical analyses indicate the superiority of the BGTOALC method in terms of efficiency and convergence rate against other binary metaheuristic algorithms.

Keywords: Feature Selection, Binary Group Teaching Optimization Algorithm, Local Search, Chaos Mapping, S-shaped and V-shaped transfer functions.

1. Introduction

High dimensional datasets usually contain extra and unrelated features that increase the complexity and reduce the accuracy of machine learning algorithms (Alweshah et al., 2020). Feature selection in datasets can considerably improve the performance of machine learning algorithms by reducing the learning model creation time and increasing the accuracy of the learning process. In recent decades, the volume of data with multiple features has been increased in many different fields, creating various challenges, such as over-fitting, high computational time, accuracy reduction, etc. (Sihwail, Omar, Ariffin, & Tubishat, 2020), in using machine learning algorithms to analyze these datasets. The quality of input features directly affects the learning model, the creation of an optimized model, and the prediction of machine learning methods. Hence, removing the extra and unrelated features brings the benefit of improvement in the accuracy and speed of the learning algorithms (Chandrashekar & Sahin, 2014). Two different approaches are designed to mitigate the mentioned challenges, comprising feature transformation and feature reduction (feature extraction) (Ding, Zhou, & Bi, 2020; Gonzalez-Lopez, Ventura, & Cano, 2020).

The feature extraction approach tries to reduce the dimensionality of the data by which an initial set of raw data is reduced to more manageable groups for processing. On the other hand, feature selection is a process of finding an essential subset of the original features, implementing no changes to the data (Abualigah, 2019).

In other words, feature selection methods remove noisy or irrelevant features from a given dataset to reach a subset of important features. This process consists of three stages: generation, evaluation, and validation. Generating the feature subset is the most crucial stage in feature selection methods, and it is considered an NP-hard optimization problem (Khaire & Dhanalakshmi, 2019). Feature selection methods are different in the evaluation technique for the candidate feature subset, so they are divided into three main groups, comprising wrapper, embodied, and filter methods (Chandrashekar & Sahin, 2014; de Souza, de Macedo, dos Santos Coelho, Pierezan, & Mariani, 2020). Filter methods are considered the fastest feature selection method, for they do not involve training the models and are not computationally expensive as well (de Souza et al., 2020). However, filter methods might fail to find the best subset of features on many occasions. On the other hand, wrapper methods use classification algorithms to measure the usefulness of a feature subset by actually applying the classifier to it, so it achieves higher accuracy in providing the best subset of features in most cases. However, they are slower than filter methods as their feature subset quality is determined by a classifier. Moreover, the embodied methods' modeling performance is weaker than filter methods, but their computation time is better than the wrapper methods (Mafarja, Heidari, Faris, Mirjalili, & Aljarah, 2020). In general, wrapper methods can yield a better subset of classified features than filter and embodied methods (Abdel-Basset, El-Shahat, El-henawy, de Albuquerque, & Mirjalili, 2020). Another approach for selecting the most important features is using metaheuristic algorithms. Metaheuristics are one of the optimization methods that are generally divided into population-based and single-solution. In population-based methods, several optimal solutions are generated simultaneously, and the operators of the desired algorithm are applied to it. Population-based algorithms are usually inspired by how animals behave or how they hunt in nature. Algorithms such as particle swarms optimization(PSO) (Poli, Kennedy & Blackwell, 2007), bat algorithm(BA) (Yang & Hossein Gandomi, 2012), grey wolf optimization algorithm(GWO) (Mirjalili, Mirjalili & Lewis, 2014), monarch butterfly optimization (MBO) (Wang, Deb & Cui, 2019), butterfly optimization algorithm(BOA) (Arora & Singh, 2019), earthworm optimization algorithm (EWA) (Wang, Deb & Coelho, 2018), elephant herding optimization (EHO) (Wang, Deb & Coelho, 2015), moth search (MS) algorithm (Wang, 2018), Slime mould algorithm (SMA) (Li, Chen, Wang, Heidari & Mirjalili, 2020), hunger games search (HGS) (Yang, Chen, Heidari & Gandomi, 2021), colony predation algorithm (CPA) (Tu, Chen, Wang & Gandomi, 2021), harris hawks optimization (HHO) (Heidari et al., 2019), and farmland fertility algorithm(FFA) (Shayanfar & Gharehchopogh, 2018) are some examples of metaheuristic algorithms.

Recently, researchers have shown an increased interest in hiring the metaheuristic methods, such as evolutionary algorithms and swarm intelligence algorithms, as wrapper-based methods in the feature selection domain for finding the most important feature subset maximizing the classification accuracy while minimizing the number of selected features (Abd Elaziz, Ewees, Ibrahim, & Lu, 2020). In this respect, various metaheuristics, such as binary particle swarms optimization (BPSO) (Mirjalili & Lewis, 2013), binary bat algorithm (BBA) (Mirjalili, Mirjalili, & Yang, 2014), binary butterfly optimization algorithm (BBOA) (Arora & Anand, 2019), binary grey wolf optimization algorithm (BGWO) (Hu, Pan, & Chu, 2020), and binary salp swarm algorithm (BSSA) (Faris et al., 2018; Tubishat et al., 2021), have been utilized to solve the above-mentioned NP-hard problem in a reasonable time with low computational cost by exploring the search space intelligently. In addition, different techniques have been used to present a metaheuristics binary algorithm in the field of feature selection, such as the S-shaped and V-shaped transfer functions for the binarization process, which transforms the continuous space to a binary one, evolutionary operators (i.e., selection, crossover, and mutation) (Mafarja et al., 2019; Yudong Zhang, Wang, Phillips, & Ji, 2014), new binary operators (Ouadfel & Abd Elaziz, 2020; Tubishat et al., 2021), etc., have been used to improve the exploration and exploitation capabilities of the metaheuristics in feature selection problems. Nevertheless, the majority of the previously developed metaheuristic algorithms are vulnerable to trap in poor local minima or have premature convergence in feature selection. Furthermore, some studies utilized chaotic functions to increase the exploitation of metaheuristic algorithms in feature selection problems (Sayed, Hassanien, & Azar, 2019; X. Zhang et al., 2020). In addition, several studies used evolutionary operators, such as mutation and crossover, to enhance the exploration and exploitation capabilities of the metaheuristics because the application of transfer functions cannot bring the benefit of high exploration and exploitation (Emary, Zawbaa, & Hassanien, 2016b; Taradeh et al., 2019). However, using mutation and crossover operators instead of the primary operators of the metaheuristic algorithm may be able to guarantee the higher exploration and exploitation partially. Although metaheuristics have been widely used on the wrapper-based feature selection methods, far too little attention has been paid to the principal process of these binary metaheuristic algorithms. That is, very few studies, such as (Santana Jr, Macedo, Siqueira, Gokhale, & Bastos-Filho, 2019), have sought to suggest a novel metaheuristic algorithm or make a significant modification in the algorithm, for they require a complete change in the main process of the algorithm and a proper design of new operators. Regarding the previous methods, the S-shaped and V-shaped transfer functions can be implemented simply and without complexity. But the problem with transfer functions is that they do not guarantee exploration and exploitation. To be percise, they have more problems with exploration and generating new solutions. Furthermore, changes in the structure of metaheuristic algorithms should be in a way that they can ensure the exploration and exploitation capabilities. However, some studies, such as (Yudong Zhang, Wang, Phillips, & Ji, 2014) study, employed the mutation operator in their methods to increase the exploration, but their methods do not have any operator for the exploitation. As a result, some researchers use crossover and mutation operators to guarantee the exploration and exploitation capabilities, yet setting the optimal value of which is a challenge. Therefore, to deal with these problems and avoid the local minima and better convergence of metaheuristic algorithms in the field of feature selection, new operators should be created according to the algorithm structure and also local search operators to increase convergence and quality of solutions.

It also should be considered that parameterizing the metaheuristic algorithm for the balance between exploration and exploitation is a major challenge. For instance, adjusting mutation and crossover operators in the genetic algorithm, the values of parameters C1 and C2 in the particle swarm algorithm can play an important role in the output. In the GTOA optimization algorithm, however, no parameter adjustment is required, and the amount of exploration and exploitation is done by dividing the population into two categories and four operators. Therefore, a remarkable aspect of the GTOA is that it relies only on population size and the number of iterations for optimization. Since the operators of this algorithm were designed continuously,

there are different solutions to use this algorithm for feature selection problems. The GTOA algorithm also has advantages over the GWO, DE, PSO, and other well-known metaheuristic algorithms in terms of convergence speed and no need to adjust parameters (Jiang, Wu, Zhang, Zhu & Xing, 2021). In addition, in solving the binary problem, if the relationship between the outstanding group and the average group is not established properly, the algorithm may get stuck in the local optima or suffer from premature convergence. Considering the superiorities of the GTOA and trying to rectify its possible drawbacks, we are motivated to present three different approaches for feature selection based on the algorithm. Two approaches called BGTOAV and BGTOAS are designed based on S-shaped and V-shaped functions to examine the GTOA without any change in the structure of the algorithm. In fact, in these two versions, the GTOA may undergo premature convergence, poor exploration, and local optima due to its structure. Therefore, to achieve a better feature selection algorithm based on the GTOA, changes in the structure of the algorithm and the use of operators guaranteeing the exploration and efficiency are required. In this paper, a new wrapper feature selection method based on the BGTOA using local search and chaos mapping to solve feature selection problems in high-dimensional data has been suggested. The proposed model uses the opposite-based mutations to increase exploration, new operators in various phases to increase exploitation and exploration, as well as the chaotic function to generate a better initial population and exploitation. In summary, the main contributions of this study can be summarized as follows.

1. We propose a novel BGTOA based on local search and chaos mapping named BGTOALC for solving high-dimensional feature selection problems.
2. We also introduce two new BTPGG and BTPBG operators for the teacher phase of the BGTOALC to increase exploration and exploitation.
3. Besides, we design the student phase of BGTOALC in a binary manner by developing a new binary operator (BSOBL), which is based on an opposition-based strategy, to increase population diversity and improve the exploitation.
4. Furthermore, the teacher allocation phase of BGTOALC is designed in a binary nature using an MBS operator to increase the algorithm's convergence rate.
5. To analyze the BGTOA performance based on transfer functions, two other approaches, i.e., BGTOAV and BGTOAS, are developed based on the S-shaped and V-shaped functions.
6. Eventually, to show the performance comparison of the proposed methods, the approaches' performance is analyzed on 30 datasets with different dimensions. Additionally, to validate the performance of the proposed methods, we conduct a comprehensive comparison between them and other well-known binary algorithms, such as BBA, BFPA, BCCSA, BGSA, BPSO, BGWO, and binary multi-neighborhood ABC (BMNABC). Different tests prove that the BGTOALC method outperforms other methods in terms of selected features number and classifier accuracy. Also, statistical analyses indicate the superior efficiency and convergence rate of the BGTOALC method compared to other binary metaheuristics.

The rest of the paper is organized as follows. The literature review of applying binary metaheuristics in the wrapper feature selection method is surveyed in the following Section 2. Section 3 represents the preliminaries, definitions, and steps of GTOA. Section 4 presents the proposed BGTOALC approach in detail. Section 5 presents simulation, computational results, and the comparison with other related works. Finally, Section 6 provides concluding remarks and suggestions for future research.

2. Literature Review

In recent years, a great deal of studies has grown up around the theme of using binary metaheuristic algorithms in feature selection methods. Table 1 lists a variety of studies on feature selection methods with metaheuristics and provides a comprehensive comparison of the previous studies based on the features, such as the base

algorithm (BaA), a dataset with the minimum number of features (MiF), a dataset with the maximum number of features (MxF), the number of utilized datasets for analyzing the proposed method (ND), utilizing V-shaped transfer functions (Vsh), utilizing S-shaped transfer functions (Ssh), applying mutation operator (Mu), applying crossover operators (Cr), using chaos mapping (Ch), proposing new operators (Nopr).

Table 1 demonstrates that in 2013, Mirjalili et al. introduced new transfer functions named S-shaped and V-shaped for BPSO (Mirjalili & Lewis, 2013). They employed twenty-five benchmark functions to evaluate the suggested transfer functions in terms of convergence rate and local minima avoidance. Afterward, in 2014, the BBA (Mirjalili et al., 2014) utilizing different S-shaped and V-shaped transfer functions were proposed. During the same year, a binary version of the PSO algorithm (Yudong Zhang et al., 2014) with the mutation operator to solve the feature selection problem was presented. In this study, the proposed MBPSO algorithm was used to detect spam emails. They conducted various experiments, and the results demonstrated that the MBPSO performed better compared to other metaheuristics, such as genetic algorithm (GA).

Table 1: A Summary of Prior Studies on Feature Selection Using Different Metaheuristic Algorithms

Ref.	BaA	MiF	MxF	ND	Vsh	Ssh	Mu	Cr	Ch	Nopr
(Mirjalili & Lewis, 2013)	BPSO	Bf	Bf	25	■	■	-	-	-	-
(Yudong Zhang et al., 2014)	MBPSO	57	57	1	■	■	■	-	-	-
(Mirjalili, Mirjalili, & Yang, 2014)	BBA	Bf	Bf	22	■	■	-	-	-	-
(Rodrigues, Yang, De Souza, & Papa, 2015)	BFPA	8	22283	6	-	■	-	-	-	-
(Emary, Zawbaa, & Hassanien, 2016b)	BGWO	9	325	18	-	■	-	■	-	-
(Emary, Zawbaa, & Hassanien, 2016a)	BALO	9	325	21	■	■	-	-	-	-
(Pashaei & Aydin, 2017)	BBHA	13	24481	8	-	■	-	-	-	-
(Faris et al., 2018)	BSSA	9	7129	22	■	■	-	-	-	-
(De Souza, dos Santos Coelho, De Macedo, & Pierezan, 2018)	BCSA	13	34	5	■	-	-	-	-	-
(Mafarja & Mirjalili, 2018)	BWOA	7129	7129	20	-	-	■	■	-	-
(Papa, Rosa, de Souza, & Afonso, 2018)	BBSO	8	10000	26	■	-	-	-	-	-

(Khuat & Le, 2019)	BTLBO	30	30	1	-	-	-	-	-	-
(Arora & Anand, 2019)	BBOA	9	325	21	■	■	-	-	-	-
(Alweshah, Khalaileh, et al., 2020)	MBO	9	325	18	-	-	-	-	-	-
(Mafarja et al., 2019)	BGOA	9	325	25	■	■	■	■	-	-
(Hussien, Hassanien, Houssein, Bhattacharyya, & Amin, 2019)	BWOA	9	325	21	-	■	-	-	-	-
(Sayed, Hassanien, & Azar, 2019)	BCCSA	8	82	20	-	-	-	-	■	-
(Santana Jr, Macedo, Siqueira, Gokhale, & Bastos-Filho, 2019)	BABC	13	166	7	-	-	-	-	-	■
(X. Zhang et al., 2020)	MCFOA	Bf	Bf	23	-	-	■	-	■	-
(Taradeh et al., 2019)	BGSA	9	325	18	-	-	■	■	-	-
(Mafarja, Heidari, et al., 2020)	BDA	7	7070	21	■	-	-	-	-	-
(de Souza et al., 2020)	BCOA	Bf	Bf	29	-	■	-	-	-	-
(Hu et al., 2020)	BGWO	9	325	18	■	■	-	-	-	-
(Mafarja, Qasem, et al., 2020)	BGWO-BWOA	Bf	Bf	23	-	■	-	-	-	-
(Kumar & Kaur, 2020)	BSHO	2322	3657	3	■	-	-	-	-	-
(Thaher, Heidari, Mafarja, Dong, & Mirjalili, 2020)	BHHO	6	22	3	■	■	-	-	-	-
(Emine & Ülker, 2020)	BinSSA	9	856	21	■	■	-	-	-	-
(Tubishat et al., 2021)	DSSA	13	1498	23	-	-	-	-	■	■
(Neggaz, Houssein, & Hussain, 2020)	BHGSO	13	15009	18	-	-	-	-	-	-

(Chaudhuri & Sahu, 2021)	BCSA	3	33	10	■	■	-	-	-	-
(Ouadfel & Abd Elaziz, 2020)	BCSA	9	325	16	-	-	-	-	-	■
(Wang, Khishe, Kaveh, & Mohammadi, 2021)	BChOA	Bf	Bf	43	■	■	-	-	-	■
Proposed	**BGTOALC**	6	19993	30	■	■	■	■	■	■

Bf = Benchmark function.

In 2015, a binary version of the Flower Pollination Algorithm (FPA) called BFPA for feature selection was suggested (Rodrigues, Yang, De Souza, & Papa, 2015). The authors used the S-shaped transfer function to develop the BFPA algorithm. Also, They evaluated the BFPA on several datasets, and the results demonstrated the suitability of the BFPA for feature selection compared to other metaheuristic-based methods. In 2016, Emary et al. proposed two BGWO algorithms using the sigmoidal function as an S-shaped transfer function and the stochastic crossover operator for the feature selection problem (Emary, Zawbaa, & Hassanien, 2016b). The authors tested the performance of the approaches on 18 datasets and showed their superiority compared to BPSO and GA algorithms. Besides, Emary et al. suggested three binary ant lion optimizers (BALO) utilizing S-shaped and V-shaped transfer functions and the crossover operator for the feature selection domain to explore a feature subset that maximizes the classification accuracy while minimizes the selected features number (Emary, Zawbaa, & Hassanien, 2016a). In 2017, Pashaei et al. proposed a binary version of the black hole algorithm (BBHA) utilizing the Hyperbolic Tangent function for feature selection in the biological datasets (Pashaei & Aydin, 2017). One year later, Souza et al. Proposed a new wrapper-based method using the V-shaped transfer function and the CSA (De Souza, dos Santos Coelho, De Macedo, & Pierezan, 2018). Moreover, Faris et al. utilized eight S-shaped and V-shaped transfer functions to propose BSSA (Faris et al., 2018). The authors also employed the crossover operators to BSSA for enhancing the exploration capability of the algorithm. They demonstrated the superiority of the proposed approaches compared to other metaheuristic algorithms, including BGWO, GA, and BPSO, on 22 UCI datasets. The study by Mafarja and Mirjalili developed new wrapper-based feature selection approaches based on the BWOA algorithm using mutation operator, crossover operator, tournament selection strategy, and roulette wheel selection strategy (Mafarja & Mirjalili, 2018). The authors implemented the proposed approaches on the UCI dataset and proved their higher efficiency than previous approaches. In the same year, Papa et al. introduced a binary brain storm optimization (BBSO) for feature selection purposes (Papa, Rosa, de Souza, & Afonso, 2018). The authors used various V-shaped transfer functions to map the continuous solutions onto a discrete one. In 2019, Khuat et al. proposed a software defect prediction method in which the authors developed a binary teaching-learning-based optimization algorithm (BTLBO) with a distribution-based solution update rule to generate an optimal sample subset for the training dataset, used by a classifier to predict potential defects (Khuat & Le, 2019). In a research study conducted by Hussien et al., a BWOA based on a sigmoid transfer function (S-shaped) was proposed to select the optimal feature subset for the classifications problem (Hussien, Hassanien, Houssein, Bhattacharyya, & Amin, 2019). In another study, a binary chaotic crow search algorithm called BCCSA was presented for the feature selection problem (Sayed, Hassanien, & Azar, 2019). In this paper, ten chaotic and S-shape transfer functions were used to improve the CSA and develop binary CSA versions. The authors showed that their method had better performance compared to other metaheuristic-based methods, such as the BPSO and BGWO. In addition, Mafarja et al. conducted a research study in which three binary grasshopper optimization

algorithms (BGOA) were proposed for the feature selection problem (Mafarja et al., 2019). The Authors employed the S-shaped and V-shaped transfer functions and the mutation operator in their suggested algorithms. The experiment results showed that exploiting the mutation operator in BGOA leads to better performance than other metaheuristics, such as BGWO, GA, and BPSO. In a similar research study, Taradeh et al. introduced a new wrapper-based feature selection approach based on the gravitational search algorithm (BGSA) with mutation and crossover operators (Taradeh et al., 2019). The authors used mutation and crossover operators to enhance the convergence rate and exploration and exploitation behaviors of the BGSA. The experiment results demonstrated that the BGSA approach performed better than other algorithms, such as GA, BPSO, and BGWO. In another study, Arora and Anand proposed two BBOA using the V-shaped and S-shaped functions for the wrapper-based feature selection method (Arora & Anand, 2019). The authors designed various experiments on 21 UCI datasets, and the results showed that the S-BBOA approach performed better than other previous algorithms. Two similar MBO algorithm for feature selection are proposed by Alweshah et al. (Alweshah et al., 2020). In this paper, the simple version of the MBO algorithm is combined with the KNN classifier algorithm. Also, a research study conducted by Santana et al. introduced a new BABC for feature selection problems (Santana Jr, Macedo, Siqueira, Gokhale, & Bastos-Filho, 2019). The author replaced arithmetic operations with flip operations over the binary vectors and an adaptable mechanism, limiting the maximum number of dimension changes. The suggested BABC algorithm was compared to other versions of the ABC algorithm, and the results indicated the superiority of the approach. Then, in 2020, Thaher et al. introduced a binary harris hawks optimizer (BHHO) based on the S-shaped and V-shaped transfer functions to enhance the effectiveness of wrapper-based feature selection methods (Thaher, Heidari, Mafarja, Dong, & Mirjalili, 2020). The authors compared the proposed approaches with other metaheuristics-based feature selection methods on high-dimensional low-sample datasets, and the results indicated the superiority of their approaches. Another study by Kumar and Kaur introduced a binary version of spotted hyena optimizer (BSHO) using the Hyperbolic Tangent function for feature selection in the medical dataset (Kumar & Kaur, 2020). The authors evaluated the BSHO performance on twenty-nine benchmark functions; then, they compared the BSHO feature selection performance with other metaheuristics over eleven UCI repository datasets, and the results showed that the BSHO algorithm achieved the optimal feature subset. Also, Souza et al. suggested a binary version of the coyote optimization algorithm (BCOA) using the Hyperbolic Tangent function for the wrapper-based feature selection method (de Souza et al., 2020). In a recent study by Mafarja et al., a hybrid metaheuristic approach using BGWO and BWOA was proposed for the wrapper-based feature selection method (Mafarja, Qasem, et al., 2020). The primary purpose of their approach was to alleviate the drawbacks of both algorithms. The results showed the superiority of their approach compared to both BGWO, BWOA, and other metaheuristic-based approaches. In another study by Hu et al., feature selection approaches based on an enhanced BGWO algorithm were proposed (Hu et al., 2020). The authors utilized five different transfer functions and also introduced a new equation to improve the BGWO by balancing the abilities of global search and local search. Moreover, Emine and Ülker published a paper in which they proposed a binary social spider algorithm (BinSSA) using the S-shaped and V-shaped transfer functions (Emine & Ülker, 2020). The authors also added a crossover operator to the BinSSA for improving the exploration and exploitation capabilities. Finally, they indicated the superior performance of the BinSSA as a wrapper-based feature selection over previous metaheuristic-based methods on 21 UCI datasets. A study by Neggaz et al. suggested a wrapper-based feature selection method based on the binary variant of the henry gas solubility optimization (BHGSO) algorithm (Neggaz, Houssein, & Hussain, 2020). The authors implemented the method on the UCI dataset, and the obtained results showed the acceptable performance of the HGSO in terms of accuracy and feature number. According to a research study by Elaziz and Ouadfel, an enhanced BCSA as a wrapper-based feature selection method was proposed (Ouadfel & Abd Elaziz, 2020). The authors used the dynamic local neighborhood and adaptive awareness probability to improve BCSA to tackle the premature convergence, leading to a reasonable

balance between exploration and exploitation. In another study, Mafarja et al. designed a wrapper-based feature selection algorithm utilizing a binary variant of the dragonfly algorithm (BDA) (Mafarja, Heidari, et al., 2020). The authors evaluated the performance of BDA on datasets containing a massive number of features with low samples, and the results indicated the superiority of the BDA over other metaheuristic-based methods. Moreover, Zhang et al. proposed a wrapper-based feature selection method based on the fruit fly optimization algorithm (FOA) with Gaussian mutational chaotic named MCFOA for the feature selection problem (X. Zhang et al., 2020). The authors developed a Gaussian mutation operator to address the premature convergence issue and a chaotic local search algorithm to enhance the exploitation ability. The experiment results indicated that the MCFOA improves the classification accuracy and reduces the number of features more efficiently. In 2021, Tubishat et al. designed a dynamic approach of the SSA algorithm (DSSA) for the feature selection problem. The authors developed a local search algorithm to improve the exploitation behavior and a new update strategy to increase the solutions' diversity (Tubishat et al., 2021). In another study, Chaudhuri et al. proposed eight variants of BCSA with time-varying flight length using S-shaped and V-shaped transfer functions for feature selection problems in wrapper mode (Chaudhuri & Sahu, 2021). Most recently, Wang et al. suggested a new binary version of the Chimp optimization algorithm (BChOA) based on the S-shaped and V-shaped functions as a wrapper-based feature selection method (Wang, Khishe, Kaveh, & Mohammadi, 2021).

3. Group Teaching Optimization Algorithm

GTOA is one of the new metaheuristic algorithms presented in 2020 by Zhang and Jin, which is inspired by group teaching (excellent group and average group) (Yiying Zhang & Jin, 2020). This algorithm was designed and implemented based on Confucius' idea of "teaching students according to their aptitude". In other words, this idea expresses a method of teaching in which the teacher is tasked with editing their teachings based on student attributes and intelligence level. In general, the GTOA is after improving the whole class's knowledge while simulating group teaching mechanisms. This algorithm follows these four rules:

1. The ability to accept knowledge is the only difference among students. The higher levels of difference between student reception ability increase the teacher's challenge in the teaching plan.
2. A good teacher pays more attention to students with weaker reception abilities than stronger students.
3. A student can gain more knowledge by self-learning and interacting with other students during their free time.
4. A good teacher allocation mechanism is important for student knowledge increase.

The GTOA algorithm consists of four phases, which are teacher allocation, ability grouping, teacher and student stages. In the following, the four phases are explained alongside these four rules (Yiying Zhang & Jin, 2020).

3.1. Ability grouping phases

All students are divided into two smaller groups in the GTOA to better show the group teaching feature without losing the possibility of generalization. These two groups are of identical importance in GTOA. The group with the higher knowledge reception ability is known as the excellent group, while the other groups with weaker reception abilities are known as the average group. According to the first rule, teachers will have an easier time using the ability grouping method instead of traditional teaching methods. The standard deviation of these two groups might increase by performing teaching activities. To solve this problem, the ability grouping is defined as a dynamic process in the GTOA, which is repeated after each learning cycle (Yiying Zhang & Jin, 2020).

3.2. Teacher phase

The teacher phase is where a student learns from their teacher (defined according to the second rule). The teacher makes different teaching plans for the excellent and average GTOA groups.

3.2.1 Teacher phase for Excellent Students

The teacher focuses on improving the knowledge of this group as a principal of the GTOA due to their high knowledge reception ability. In other words, the teacher does their best in improving the class average while considering the different student knowledge reception levels. Therefore, the excellent group of students can gain their knowledge as follows (Yiying Zhang & Jin, 2020).

$$X_{teacher,i}^{t+1} = X_i^t + a \times \left(T^t - F\left(b \times M^t + c \times X_i^t\right)\right) \quad (1)$$

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

$$b + c = 1 \quad (3)$$

In equations. (1)-(3), t is the current iteration, N is the number of students, x_i^t is knowledge of student i in t time, T^t is the teacher's knowledge at t time, M^t is the average group knowledge at time t, F is the teaching factor that determines teaching results, $x_{teacher,i}^{t+1}$ is the knowledge of student i at t time learning from the teacher while a, b, and c are random numbers in the range of $[0,1]$. Note that F can be equal to 1 or 2.

3.2.2 Teacher phase for average Students

The teacher wanting to improve the personal knowledge levels in their class pays more attention to the average group compared to the excellent group (Based on rule 2) due to their weaker knowledge reception ability. Therefore, the students of this average group can gain knowledge as follows (Yiying Zhang & Jin, 2020).

$$X_{teacher,i}^{t+1} = X_i^t + 2 \times d \times (T^t - X_i^t) \quad (4)$$

In which d is a random number between $[0,1]$. A student might not gain any knowledge during the teacher phase. This fact is shown as follows (Yiying Zhang & Jin, 2020).

$$X_{teacher,i}^{t+1} = \begin{cases} X_{teacher,i}^{t+1}, & f\left(X_{teacher,i}^{t+1}\right) < f\left(X_i^t\right) \\ X_i^t, & f\left(X_{teacher,i}^{t+1}\right) \geq f\left(X_i^t\right) \end{cases} \quad (5)$$

3.3. Student phase

This phase includes Student phase one and Student phase two, corresponding to the third rule. A student can gain knowledge in their free time using two different methods: self-learning or interaction with other students, which can be expressed as follows (Yiying Zhang & Jin, 2020).

$$X_{teacher,i}^{t+1} = \begin{cases} X_{teacher,i}^{t+1} + e \times \left(X_{teacher,i}^{t+1} - X_{teacher,j}^{t+1}\right) + g \times \left(X_{teacher,i}^{t+1} - X_i^t\right), \\ f\left(X_{teacher,i}^{t+1}\right) < f\left(X_{teacher,j}^{t+1}\right) \\ X_{teacher,i}^{t+1} - e \times \left(X_{teacher,i}^{t+1} - X_{teacher,i}^{t+1}\right) + g \times \left(X_{teacher,i}^{t+1} - X_i^t\right), \\ f\left(X_{teacher,i}^{t+1}\right) \geq f\left(X_{teacher,j}^{t+1}\right) \end{cases} \quad (6)$$

In equation (6), e and g are two random numbers from $[0,1]$, $x_{student,i}^{t+1}$ is the knowledge of student i in t time of the student phase and $x_{teacher,i}^{t+1}$ is the knowledge of student j in t time of teacher learning. Regarding student j, $j \in \{1, 2, ..., i-1, i+1, ..., N\}$. This student is selected randomly. The second and third right-hand items of

equation (6) represent learning from other students and self-learning. A student might learn nothing during the student phase, which is expressed as follows (Yiying Zhang & Jin, 2020).

$$X_i^{t+1} = \begin{cases} X_{teacher,i}^{t+1}, & f(X_{teacher,i}^{t+1}) < f(X_{student,i}^{t+1}) \\ X_{teacher,i,}^{t+1} & f(X_{teacher,i}^{t+1}) \geq f(X_{student,i}^{t+11}) \end{cases} \quad (7)$$

In equation (7), x_i^{t+1} is knowledge of student i during the $t-1$ time after one learning cycle.

3.4. Teacher Allocation phase

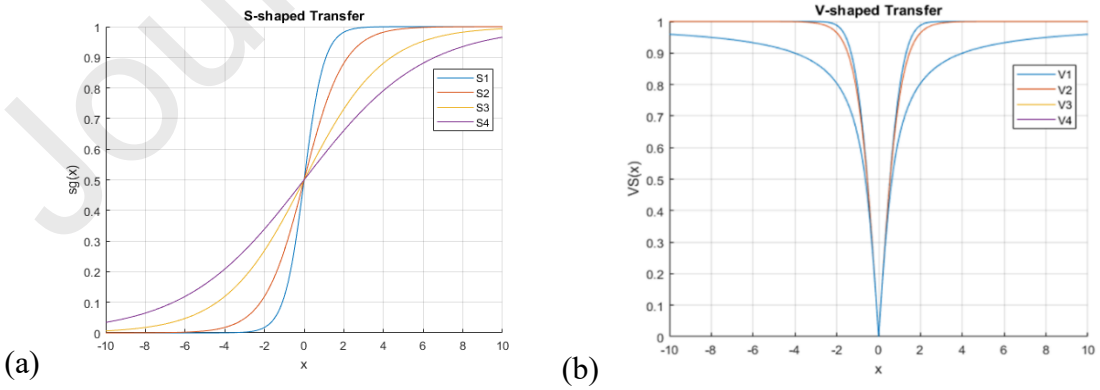
Based on the fourth rule, designing a good teacher allocation mechanism is rather important in improving student knowledge. The Gray Wolf Optimizer (GWO) saves and uses the first three solutions for guiding the wolves. The teacher allocation of the proposed method is expressed as follows with inspiration from the GWO wolf hunt behavior (Yiying Zhang & Jin, 2020).

$$T^t = \begin{cases} X_{first}^t, & f(X_{first}^t) \leq f\left(\dfrac{X_{first}^t + X_{second}^t + X_{third}^t}{3}\right) \\ \dfrac{X_{first}^t + X_{second}^t + X_{third}^t}{3}, & f(x_{first}^t) > f\left(\dfrac{X_{first}^t + X_{second}^t + X_{third}^t}{3}\right) \end{cases} \quad (8)$$

In equation (7), x_{first}^t, x_{second}^t and x_{third}^t are the first, second and third best students. The excellent and average groups share the same teacher to increase the speed of GTOA convergence.

4. Proposed Algorithm

In this section, three versions of the binary GTOA algorithm are presented. The first model uses four S-shaped transfer functions named BGTOAS1, BGTOAS2, BGTOAS3, and BGTOAS4 to present a binary GTOA. The second model uses four V-shaped transfer functions named BGTOAV1, BGTOAV2, BGTOAV3, and BGTOAV4 to present a binary GTOA. The third model implements some fundamental changes in the continuous GTOA algorithm operators to present an improved version of it named BGTOALC. The proposed BGTOALC uses local searching and chaos mapping to increase exploitation. Moreover, these two new BTPGG and BTPBG operators are designed for the teacher phase, and a new BSOBL operator is designed for the student phase. The full description of each proposed approach is presented in the following sections.



(a) (b)

Figure 1: Graphical View of Different S-shaped and V-shaped Transfer Functions

4.1. Proposed BGTOAS Algorithm

Four S-shaped transfer functions in GTOA are defined and presented in Table 2. In the BGTOAS version, the continuous X_i^{t+1} solution is transferred to a new space by each transfer function from Table 2 in the range [0,1]. Therefore, the GTOA algorithm solutions are mapped into a new space between 0 and 1. Then, randomized thresholding is used to generate a new binary solution that can be expressed in the form of equation (9).

$$X_i^{t+1} = \begin{cases} 0 & if\ r_1 < S_{model}(X_i^{t+1}) \\ 1 & if\ r_1 \geq S_{model}(X_i^{t+1}) \end{cases} \quad (9)$$

where r_1 is a number in the range [0,1] that determines the final value of each of the dimensions of X_i^{t+1}, thereby generating a new binary solution. Besides, X_i^{t+1} denotes the position of i solution during iteration t of the GTOA. S_{model} determines each BGTOAS1, BGTOAS2, BGTOAS3, and BGTOAS4 transfer function shown in Table 2, as well as the graphical representation of all four S-shaped transfer functions illustrated in Figure 1(a).

Table 2: Various S-shaped and V-shaped Transfer Functions in BGTOAS

Method name	Transfer function	Types
BGTOAS1	$sg(X_i^{t+1}) = \frac{1}{1+e^{-2X_i^{t+1}}}$	S-shaped
BGTOAS2	$sg(X_i^{t+1}) = \frac{1}{1+e^{-X_i^{t+1}}}$	
BGTOAS3	$sg(X_i^{t+1}) = \frac{1}{1+e^{(-\frac{X_i^{t+1}}{2})}}$	
BGTOAS4	$sg(X_i^{t+1}) = \frac{1}{1+e^{(-\frac{X_i^{t+1}}{3})}}$	
BGTOAV1	$VS(X_i^{t+1}) = \left\lvert \text{erf}\,(\frac{\sqrt{\pi}}{2}X_i^{t+1}) \right\rvert$	*V-shaped*
BGTOAV2	$VS(X_i^{t+1}) = \lvert \tanh\,(X_i^{t+1}) \rvert$	
BGTOAV3	$VS(X_i^{t+1}) = \left\lvert \frac{X_i^{t+1}}{\sqrt{1+X_i^{t+1^2}}} \right\rvert$	
BGTOAV4	$VS(X_i^{t+1}) = \left\lvert \frac{2}{\pi} arc\tan\,(\frac{\pi}{2}X_i^{t+1}) \right\rvert$	

Figure 2 shows The BGTOAS algorithm pseudo-code. The different BGTOAS1, BGTOAS2, BGTOAS3, and BGTOAS4 transfer functions are implemented in the BGTOAS algorithm, and then each equation is turned into a binary one using equation 9.

```
01: Define fitness function using Equation (12).
02: Initial parameter
03: Initialize the population
04: Calculate the fitness function of the initial solution f(X_i)
05: Sort and get the first, second, third solution
06: while t ≤ Max_iteration do
07:     The teacher is selected by Equation (8).
08:     Convert solution to new space using S_model.
09:     Convert solution to binary using Equation(9).
```

10: The best half individuals form the outstanding group**.**
11: The rest individuals make up the average group.
12: ***// Step : Outstanding Group***
13: Perform the teacher phase using Equations 1 to 3 and 5.
14: Perform the student phase using Equations 6 to 7.
15: Convert solution to new space using S_{model}.
16: Convert solution to binary using Equation (9).
17: The new population of the outstanding group.
18: ***// Step: Average Group***
19: Perform the teacher phase using Equations (4) to (5).
20: Perform the student phase using Equations (6) to (7).
21: Convert solution to new space using S_{model} in Table (1).
22: Convert solution to binary using Equation (9).
23: The new population of the average group.
24: ***// Step: Average Group***
25: Construct a new population.
26: Select the optimum solution.
27: **end while**

Figure 2: Proposed BGTOAS Pseudo-code

4.2. Proposed BGTOAV Algorithm

This section implements four V-shaped transfer functions in the GTOA, as depicted in Table 2. In the BGTOAV version, the continuous X_i^{t+1} solution is transferred to a new space by each V-shaped transfer function from Table 2 in the range [0,1]. Therefore, the GTOA solutions are mapped into a new space between zero and one. Randomized thresholding is then used to generate a new binary solution that can be expressed in the form of equation (10).

$$X_i^{t+1} = \begin{cases} 0 & if\ r_1 < V_{model}\left(X_i^{t+1}\right) \\ 1 & if\ r_1 \geq V_{model}\left(X_i^{t+1}\right) \end{cases} \quad (10)$$

where r_1 is a number in the range [0,1] that determines the final dimensions of the X_i^{t+1}, creating a new binary solution. Also, X_i^{t+1} is the i solution crowd position during the t iteration of the GTOA. The V_{model} determines each BGTOAV1, BGTOAV2, BGTOAV3, and BGTOAV4 transfer function shown in Table 2 alongside the graphical representation of all four V-shaped transfer functions in Figure 1(b).

01: Define fitness function according to using Equation (12).
02: Initial parameter
03: Initialize the population
04: Calculate the fitness function of the initial solution $f(X_i)$
05: Sort and get the first, second, third solution
06: **while** t $\leq Max_{iteration}$ **do**
07: The teacher is selected by Equation (8).
08: Convert solution to new space using V_{model} .
09: Convert solution to binary using Equation(10).
10: The best half individuals form the outstanding group**.**
11: The rest individuals make up the average group.
12: ***// Step : Outstanding Group***
13: Perform the teacher phase using Equations 1 to 3 and 5.
14: Perform the student phase using Equations 6 to 7.
15: Convert solution to new space using V_{model}
16: Convert solution to binary using Equation (10).
17: The new population of the outstanding group.
18: ***// Step: Average Group***
19: Perform the teacher phase using Equations (4) to (5).
20: Perform the student phase using Equations (6) to (7).
21: Convert solution to new space using V_{model}.

```
22:        Convert solution to binary using Equation (10).
23:        The new population of the average group.
24:        // Step:  Average Group
25:        Construct a new population.
26:        Select the optimum solution.
27:  end while
```

Figure 3: Proposed BGTOAV Pseudo-code

Figure 3 shows The BGTOAV algorithm pseudo-code. The different BGTOAV1, BGTOAV2, BGTOAV3, and BGTOAV4 transfer functions are implemented in the BGTOAV algorithm, and then each equation is turned into a binary one using equation 10.

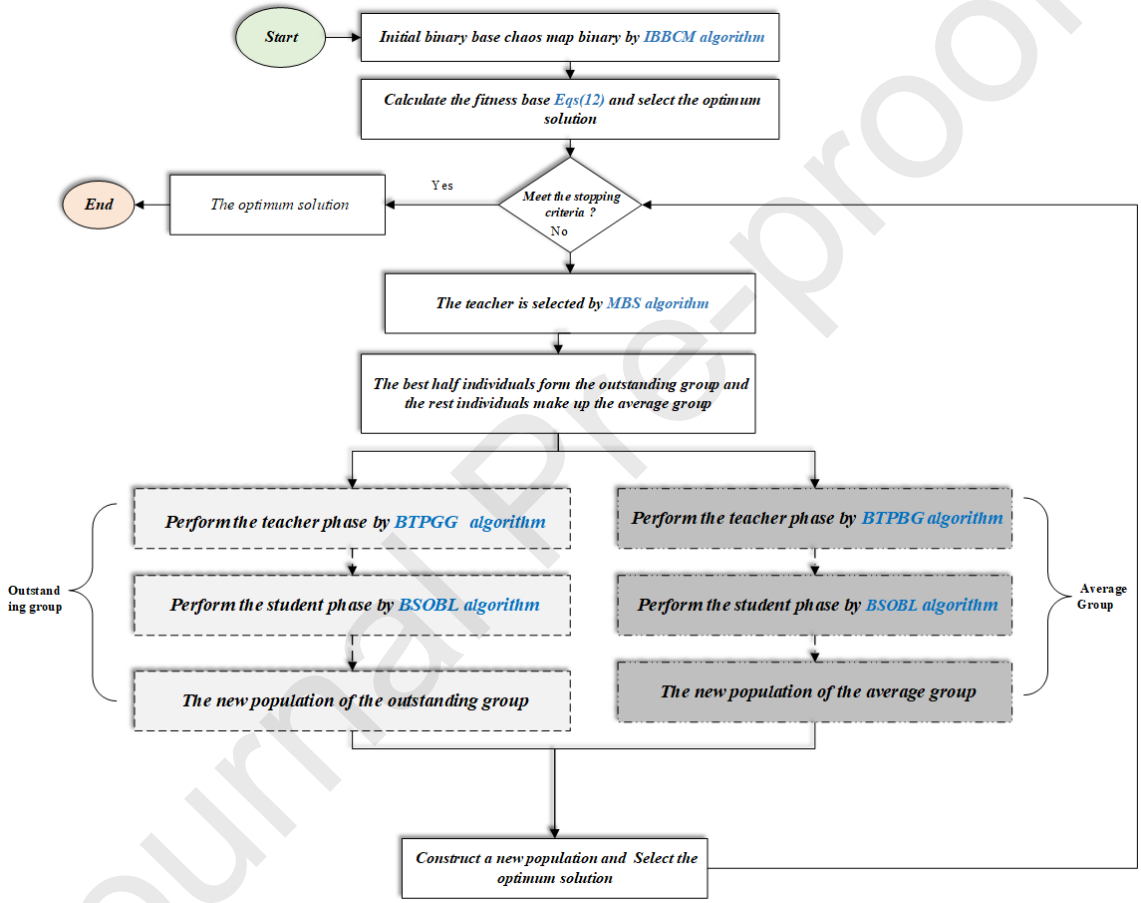


Figure 4: Proposed BGTOALC Flowchart

4.3. Proposed BGTOLAC Algorithm

In This section, a new GTOA-based wrapper method called BGTOALC is designed and implemented for feature selection and maintaining a balance between exploration and exploitation. In the BGTOALC method, significant changes are applied to the GTOA based on presenting a binary version and solving the feature selection problem. The proposed method employs a piecewise chaotic map to generate the initial binary population, which leads to a better initial population and increases the initial exploitation of the proposed BGTOALC method. The BGTOALC method has applied more changes in the teacher and student phases based on feature selection and binary [0,1] number generation to maintain a balance between exploration and exploitation of the algorithm in solving feature selection problems. The proposed BGTOALC defines the teacher phase in a binary manner using the new BTPGG and BTPBG operators while defining the student

phase in a binary manner using the new BSOBL operator. Also, the teacher allocation stage is designed in a binary manner using the new MBS operator. The steps of the proposed BGTOALC are presented as a flowchart in Figure 4.

Algorithm 1: Initial Binary Base Chaos Map (IBBCM)

01: $chaos_{rand}$ = *generate random using Piecewise chaotic map*
02: c = 1
03: **for** *i =1 to N population* ***do***
04: **for** j =1 to dimensions ***do***
05: ***if*** $(chaos_{rand}(c) < 0.5)$
06: Set $X_{(i,j)}=1$
07: ***otherwise***
08: Set $X_{(i,j)}=0$
09: ***end if***
10: Set counter c = c+ 1
11: ***end for j***
12: ***end for i***

4.3.1. Initial Binary Population generation based on Chaos Mapping

Generating an appropriate initial population significantly affects the output of the optimization problem in metaheuristic algorithms. In this paper, we face a binary problem and must generate an initial binary population of 0s and 1s, where 0 suggests that the intended feature is not selected, while 1 shows that the intended feature is selected. If a dataset has F features, the solutions to the algorithm need to be expressed as equation (11).

$$pop = \begin{bmatrix} x_{1,1}^t & x_{1,2}^t & \cdots & x_{1,F}^t \\ x_{2,1}^t & x_{2,2}^t & \ldots & x_{1,F}^t \\ \vdots & & & \vdots \\ x_{N,1}^t & x_{N,2}^t & \cdots & x_{N,F}^t \end{bmatrix} \rightarrow \begin{bmatrix} 0|1 & 0|1 & \cdots & 0|1 \\ 0|1 & 0|1 & \ldots & 0|1 \\ \ldots & & \ldots & \ldots \\ 0|1 & 0|1 & \cdots & 0|1 \end{bmatrix} \quad (11)$$

According to equation (11), the BGTOALC algorithm must generate a random initial population of 0s and 1s based on chaos mapping, and the initial population production pseudo-code is shown. Based on the algorithm pseudo-code 1, we generate a series of random numbers between 0 and 1 using a piecewise chaotic map and store them in $chaos_{rand}$. Then $chaos_{rand}$<0.5 conditions are analyzed considering the population (N) and the number of features (F) to give a value equal to 0 or 1 to each dimension of the solution to the BGTOALC. If this condition holds, the value will be equal to 1; otherwise, it will be equal to 0. The piecewise chaotic map is defined in equation (12).

$$x_{k+1} = \begin{cases} \frac{x_k}{P}, & 0 \le x_k \le P \\ \frac{x_k - 1}{0.5 - P}, & P \le x_k < 0.5 \\ \frac{1 - P - x_k}{0.5 - P}, & 0.5 \le x_k < 1 - P \\ \frac{1 - x_k}{P}, & 1 - P \le x_k < 1 \end{cases} \quad (12)$$

The piecewise chaotic map is given in full in Equation 12, where P is a constant defined between 0 and 1. It is better to set it from zero to 0.5. In this relation, the value of P is equal to 0.4, the value of $x_0$is equal to 0.7. The value of x_{k+1}also refers to the chaotic number generated by the piecewise chaotic map. In algorithm 1, this number is stored in the variable $chaos_{rand}$, so it can help the proposed algorithm to generate a better population. Since the numbers generated by the piecewise chaotic map are better than the random mode, it is suggested to generate a population with this chaotic function compared to the random mode to assist the proposed algorithm at this stage.

4.3.2. The Binary Teacher Phase with BTPGG and BTPBG Operators

The teacher phase of the continuous GTOA is defined by equations (1) and (4), in which the teacher makes different teaching plans for the average and excellent groups. In the proposed GTOA algorithm, the teacher uses X_i^t, T^t and $M^t$of equation (1) to generate a new solution for the excellent students, where b is the M^t coefficient, c is the X_i^t coefficient and a is the general coefficient used with T^t. The binary model uses the BTPGG operator for this operation; the related pseudo-code is presented in Algorithm 2.

Algorithm 2: Binary Teacher Phase Good Group(BTPGG)

01:$X_{new} =$ generate zeros 1 to dimensions
02: ***for*** n =1 to dimensions **do**
03: a= generate random 0 to 1.
04: b= generate random 0 to 1.
05: c=1-b
06: ***if*** (a > 0.7)
07: X_{new} (n) = use the teacher′s knowledge at t time→ T^t (n)
08: ***else if*** (b > c)
09: X_{new} (n) = use average group knowledge at time t→ Mt (n)
10: ***otherwise***
11: X_{new} (n) = use knowledge of student i in t time → X_i^t (n)
12: ***end if***
13: ***end for***

The new solutions for excellent students are generated using X_i^t, T^t and M^t based on the new BTPGG operator. While the aforementioned coefficients determine the degree to which X_i^t, T^t and M^t solutions are used. If the BGTOALC algorithm reaches a proper convergence, the degree to which T^t solution is used can be controlled if the $a > 0.7$ condition holds. (0.7 was selected based on different test results). Generally, each solution can inherit 30% of the optimized solution to reach convergence. Next, the average student phase is executed according to equation (4) using the BTPBG operator expressed in Algorithm 3.

Algorithm 3: Binary Teacher Phase Bad Group (BTPBG)

01: ***for*** n=1 to dimensions
02: d=2 $\times rand(0,1)$
03: ***if*** (d > 1)
04: X_{new} (n)= use the teacher′s knowledge at t time→ T^t (n)
05: ***else if*** *(d > 0.5)*
06: $X_{new}(n) =$ use knowledge of student i in t time → X_i^t (n)
07: ***else***
08: X_{new} (n) = generate random 0|1.

09: ***end if***
10: ***end for***

The new solutions for average students are generated using X_i^t and T^t based on the new BTPBG operator. The parameter $2 * d$ directly affects the solution generation method of this operator. This operator is somewhat similar to the continuous model, with the major difference that this binary version generates binary solutions. Moreover, the exploration operation of this operator is sometimes performed according to Line 8. The new BTPBG operator is graphically represented in Figure 5 to help better understand it.

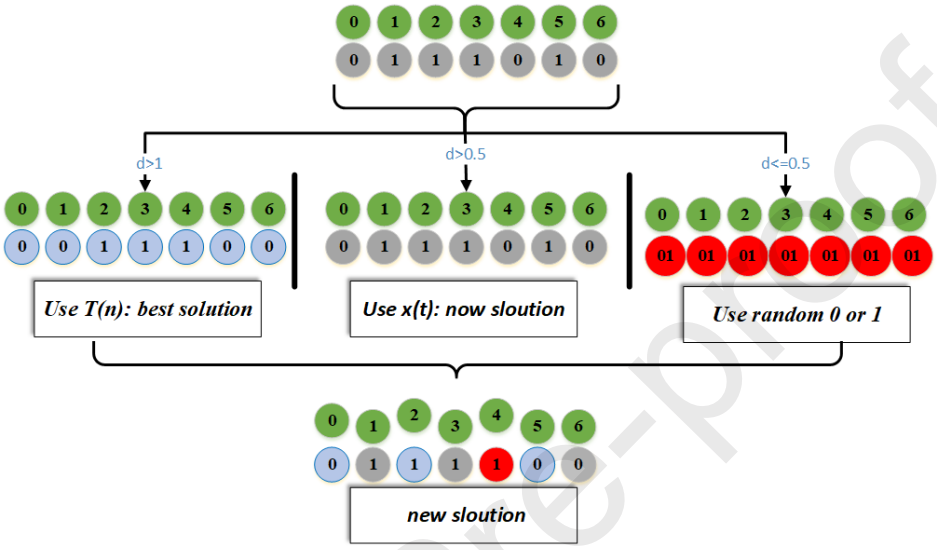


Figure 5: Binary Teacher Phase Bad Group (BTPBG) Process

4.3.3. The Binary Teacher Allocation phase with the MBS Operator

The teacher allocation phase of the continuous GTOA is defined as equation (8) inspired by the behavior of the GWO algorithm Here, x_{first}^t, x_{second}^t and x_{third}^t are considered as the first, second, and third best students. The binary version of this phase follows the MBS algorithm presented as Algorithm (4). Therefore, we count the number of 1s and 0s of each feature for the three main solutions, including x_{first}^t, x_{second}^t and x_{third}^t. If there are more 1s than 0s, the solution T^t t is filled with 1s and otherwise filled with 0s.

Algorithm 4: Mean Binary Select(MBS)

01: *Get* three solutions, including: x_{first}^t,x_{second}^t *and* x_{third}^t
02: L=length of X_1 or dimensions
03: X_{new} = generate zeros 1 to L
04: ***for*** i=1 to dimensions or L **do**
05: x=concatenate three solution: [$x_{first}^t(i)$,$x_{second}^t(i)$ *and* $x_{third}^t(i)$]
06: $count_{\text{no\_select}}$ = the sum of the cells that are zero.
07: $count_{\text{select}}$ = the sum of the cells that are ones.
08: ***if*** ($count_{\text{select}} \geq count_{\text{no\_select}}$)
09: X_{new} (i) →1 ;
10: ***else***
11: X_{new} (i) → 0 ;
12: ***end if***
13: ***end for***

According to the new MBS operator and Figure 6, the new binary solution is generated as an average based on 0s and 1s in each dimension. In effect, this operator guarantees the use of all three binary solutions x^t_{first}, x^t_{second} and x^t_{third} in generating a novel binary solution. The new MBS operator is graphically represented in Figure 6.

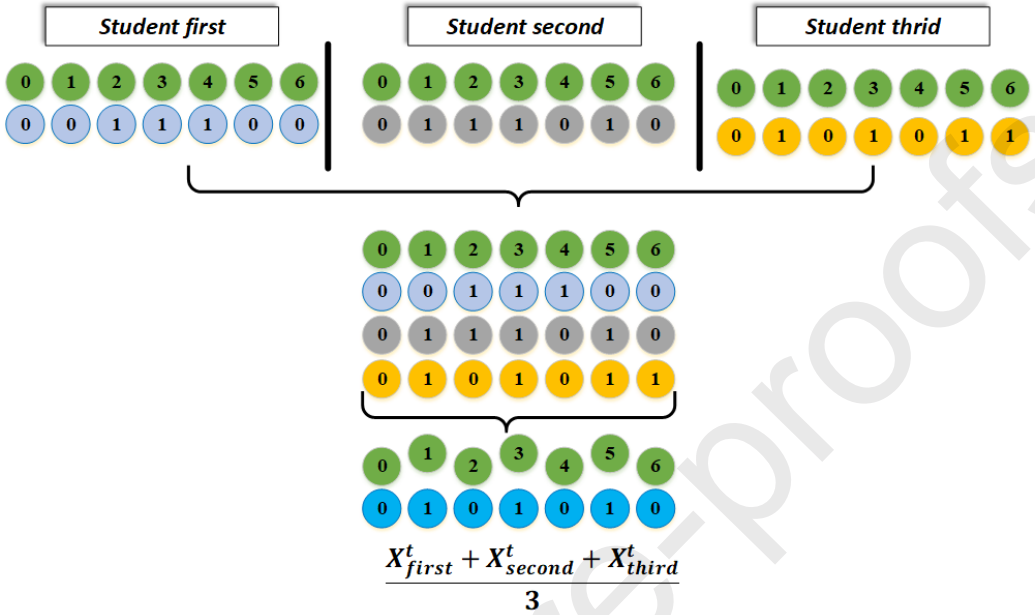


Figure 6: Mean Binary Select (MBS) Process

4.3.4. The Binary Student phase with the BSOBL Operators

The student phase of the continuous GTOA is defined as equation (6). According to the continuous GTOA, a student gains knowledge in two different ways: by self-learning or by interaction with other students. We have followed the BSOBL algorithm (Algorithm 5) in our proposed binary model. Therefore, the difference between $X^{t+1}_{teacher,i}$ and $X^{t+1}_{teacher,j}$ is calculated according to line 02. Then, we fill the stops bigger than 0 with 1 and the stops smaller than 0 with -1 while using an opposition-based local search. The opposition-based method converts 0s to 1s and vice versa in feature selection. We have used some other rules to improve the exploitation, defined as the LSBOBL algorithm (Algorithm 6).

Algorithm5: Binary Student Opposition Based Learning (BSOBL)

01: $g = rand(0,1)$
02: $point =$ Get a new solution using Equation (6).
03: if point >0, set $\rightarrow +1$;
04: if point <0, set $\rightarrow -1$;
05 :***if*** $(teacher_i\ better\ than\ teacher_j)$
06 : $X_{new} = get\ new\ position\ by\ LSBOBL(point, X_{new}, -1)$;
07: ***else***
08 : $X_{new} = get\ new\ position\ by\ LSBOBL(point, X_{new}, 1)$;
09: ***end***

The BSOBL algorithm uses the opposition-based method in Lines 6 and 8 to generate a novel solution. Here, if $f(X^{t+1}_{teacher,i}) < f(X^{t+1}_{teacher,j})$, -1 points are changed by LSBOBL; otherwise, the +1 points are changed. Therefore,

we have designed a new operator according to equation (6) which is able to change $X_{teacher,i}^{t+1}$solution using the opposition-based method.

Algorithm 6: Local Search Base Opposition Based Learning (LSBOBL)
01: Get point, x and value 02:$X_{new}\ copy\ of\ x$ 03:c=0 04:L=length(x) or dimensions 05:***for*** i=1 to L **do** 06: ***if*** (point[i]==value **and** rand (0,1) <0.5) 07: $X_{new}(i)$→1-x(i); 08: Set counter c = c+ 1 09: ***end if*** 10***: end for***

4.3.5. The Objective Function

The objective function of the feature selection problem, according to equation (13), can be a hybrid of a classifier error and a reduction of the number of the selected features. Indeed, the objective functions determine that the number of selected features must be minimized to increase the classification accuracy of the algorithm. The objective function is defined in equation (13).

$$Fitness = \alpha \times (1 - Classifier_{acc}) + \beta \times \frac{fs}{S} \quad (13)$$

where $Classifier_{acc}$ refers to the accuracy of a classification algorithm, such as the KNN. Here, α is the importance of classification accuracy, and $(1-\alpha)$ or β is the importance of selected features. Also, f_s is the number of the selected features, and s refers to all available features in a given set.

5. Results and Evaluation

The three BGTOALC, BGTOAV, and BGTOAS were implemented in MATLAB on a system with an Intel(R) Core(TM) i5-6200U CPU @ 2.40GHz and 8GB RAM. Also, the MATLAB version 2021b is used for simulating the algorithms, which is installed on Windows 10 and the 64-bit operating system.

All the three proposed methods will be analyzed using different statistical criteria. The results of their evaluation of over 30 datasets with different dimensions are presented in Table 3. The datasets are listed based on their order and contain information such as name, sample size, number of features, number of classes, area, and type of each dataset. In addition, BGTOAS1, BGTOAS2, BGTOAS3, and BGTOAS4 are compared with BGTOAV1, BGTOAV2, BGTOAV3, and BGTOAV4 to select an approach as the final transfer function-based approach.

Table 3: Full Description of the Dataset

ID	Name	Sample	Features	class	Data Area	Categories
Data1	CongressEW	434	17	2	Politics	low
Data2	Dermatology	366	35	6	Biology	medium

Data3	Diabeticretinopathy	1151	20	2	Life	medium
Data4	ORL	400	1025	40	image	high
Data5	SMK-CAN-187	187	19994	2	Microarray,Biology	high
Data6	Steel	1941	34	2	Physical	medium
Data7	abalone	4177	9	28	Life	low
Data8	audiology	199	71	24	Life	mediom
Data9	breastEW	569	31	2	Biology	low
Data10	cancerwisconsin	699	10	2	Biology	low
Data11	german	1000	25	2	Financial	mediom
Data12	glass	214	11	6	Physical	low
Data13	hepatitis	155	20	2	Life	mediom
Data14	isolet	1560	618	26	Computer	high
Data15	krvskp	3195	37	2	Game	mediom
Data16	libras	360	91	15	N/A	high
Data17	lung	203	3313	5	voice	high
Data18	lungcancer	32	57	2	Life	mediom
Data19	lymphography	148	19	4	Biology	low
Data20	lymphoma	96	4027	9	Life	high
Data21	madelon	2600	501	2	N/A	high
Data22	pdspeech	756	755	2	Computer	high
Data23	relathe	1440	1025	20	image	high
Data24	seismicbumps	2584	19	2	N/A	low
Data25	soybean	307	36	19	Life	mediom
Data26	tictactoe	958	10	2	Game	low
Data27	tox171	171	5749	4	microarray	high
Data28	warpPIE	210	2421	10	mage, face	high
Data29	wine	178	14	3	Chemistry	low
Data30	zoo	101	17	7	Artificial	low

Therefore, subsection (5-2) analyzes the effects of these eight transfer functions on the BGTOA algorithm, which results show the superiority of BGTOAS1. This leads to the selection of BGTOAS1 as the final transfer function-based approach. As mentioned earlier, several independent researchers have compared their methods with the previous well-known binary metaheuristic algorithms. Hence, in subsection (5-3), we also compares the BGTOALC method with other state-of-the-art methods, such as BPSO [25], BBA [26], BFPA [42], BGWO [38], BCCSA [36], BGSA [40], and BMNABC [57] methods. Table 4 indicates the primary parameters of the proposed BGTOALC and previous algorithms.

Table 4: The Primary Parameters of the BGTOALC and Other Comparative Algorithms

Algorithm Name	Parameter	Value
BBA	Npop	20
	Maxit	100
	A	0.5
	R	0.5
	Qmin	0
	Qmax	2
BPSO	C1	2.05
	C2	2.05
	W	2
BCCSA	Npop	20
	Maxit	100
	AP	0.1
	Fl	2
BGSA	Npop	20

	Maxit	100
	ElitistCheck	1
	min_flag	1
	Rpower	1
BGTOALC	Npop	20
	Maxit	100
BGWO	Niter	100
	Nagents	20
BMNABC	Npop	20
	Maxit	100
	Rmin	0
	Rmax	1
	Vmax	6
	Limit	10
BFPA	Npop	20
	Maxit	100
	P	0.8

5.1. Optimized Parameters α and β in the Objective Function

The proposed objective function defined in equation (13) is a hybrid of classifier error and reduction of the number of selected features. The output of this proposed algorithm and other comparative algorithms are significantly affected by parameters α and β. (set around 0.90 and 0.1, respectively, in most cases) [29]. This section compares BGTOAS1, BGTOAS2, BGTOAS3, and BGTOAS4 with BGTOAV1, BGTOAV2, BGTOAV3, and BGTOAV4 for α and β values, as well as the BGTOALC on low-, medium-, and high-dimensional datasets. Therefore, this section calculates the feature selection accuracy and reduction rate for the proposed approaches for different α and β values. The average accuracy function of the proposed methods is analyzed with different α and β values and reported in Table 5.

Table 5: Comparing the Proposed Methods in Terms of Average Accuracy with Different α and β Values

Data	BGTOAS1	BGTOAS2	BGTOAS3	BGTOAS4	BGTOAV1	BGTOAV2	BGTOAV3	BGTOAV4	BGTO
α=0.90, β=0.1									
Data1	0.854	0.30	0.111	0.301	0.214	0.18	0.435	0.397	**0.91**
Data8	0.89	0.021	0.274	0.045	0.109	0.079	0.108	0.115	**0.4**
Data27	0.279	0.105	0.10	0.244	0.14	0.279	0.279	0.179	**0.2**
α=0.95, β=0.05									
Data1	0.901	0.324	0.31	0.482	0.411	0.217	0.565	0.443	**0.9**
Data8	0.91	0.121	0.058	0.367	0.192	0.074	0.11	0.102	**0.4**
Data27	0.688	0.342	0.256	0.427	0.393	0.183	0.363	0.183	**0.5**
α=0.99, β=0.01									
Data1	0.914	0.915	0.646	0.895	0.497	0.264	0.615	0.542	**0.9**
Data8	0.939	0.834	0.932	0.938	0.204	0.074	0.12	0.102	**0.6**
Data27	0.699	0.708	0.709	0.708	0.415	0.213	0.429	0.349	**0.6**

Table 5 compares the proposed methods for average accuracy with different α and β values. Therefore, the proposed methods with $\alpha = 0.01$ and $\beta = 0.99$ achieved the best average accuracy. The results show that the higher β values considerably decrease the proposed method accuracy, and the higher α values considerably increase that value.

5.2. Analysis of S-shaped and V-shaped Transfer Function-based Methods

This section compares BGTOAS1, BGTOAS2, BGTOAS3, and BGTOAS4 with BGTOAV1, BGTOAV2, BGTOAV3, and BGTOAV4 to select a final transfer function-based approach. This test was configured with 100 iterations and population size of 20 to analyze all S-shaped and V-shaped methods based on the average number of selected features, average classification accuracy, and average objective function value. The first test analyzed BGTOAS1, BGTOAS2, BGTOAS3, and BGTOAS4 approaches as well as BGTOAV1, BGTOAV2, BGTOAV3, and BGTOAV4 based on the number of selected features. The results are presented in Table (6).

Table 6: Comparing the Proposed BGTOA Methods Based on S-shaped and V-shaped Transfer Functions in terms of the Average Number of Features

Datasets	BGTOAS1	BGTOAS2	BGTOAS3	BGTOAS4	BGTOAV1	BGTOAV2	BGTOAV3	BGTOAV4
Data1	12.3	14.65	15.24	8.37	9.39	10.68	11.65	4.38
Data2	5.75	22.39	25.91	17.05	14.45	22.51	8.37	7.33
Data3	7.32	17.49	15.87	11.96	8.06	13.27	14.68	7.25
Data4	423.2	563.81	752.3	251.05	854.36	871.34	722.98	652.74
Data5	7852	9846	5314	11258	14257	17347	16248	15379
Data6	2.0498	5.0147	3.01125	8.354	17.152	27.167	30.214	27.256
Data7	1.2145	3.125	2.145	4.258	5.214	7.259	6.325	8.214
Data8	15.214	19.856	16.982	24.658	36.247	42.354	59.25	63.248
Data9	12.302	7.321	5.0259	15.36	19.36	22.364	27.325	20.3617
Data10	2.025	1.0257	3.2458	4.34	6.214	8.3654	7.324	8.379
Data11	7.214	8.2466	12.896	14.965	17.357	18.354	22.39	19.3145
Data12	3.214	2.214	1.954	4.378	8.345	7.141	6.3124	9.214
Data13	5.324	12.14	8.325	4.2459	14.3254	11.245	12.354	18.344
Data14	214.31	324.379	125.347	382.214	523.985	553.921	465.378	429.482
Data15	9.3245	5.3217	14.361	17.395	28.752	33.612	25.937	19.765
Data16	22.39	17.349	36.89	31.96	48.652	75.32	69.489	82.854
Data17	1102.	1389.84	759.364	924.149	3001.36	2469.78	2019.75	2854.3
Data18	14.325	19.589	24.674	17.985	38.652	48.458	50.632	44.578
Data19	8.458	5.145	9.347	6.378	15.478	18.698	12.847	16.854
Data20	1402.7	1702.9	1345.6	2178.8	2963.1	3574.6	2459.6	3325.5
Data21	96.2	178.2	245.7	195.4	452.3	369.1	293.6	257.9
Data22	247.3	196.5	291.1	308.5	360.5	693.48	514.82	701.6
Data23	395.1	469.5	214.9	512.6	792.5	951.4	824.6	746.3
Data24	2.36	9.34	5.78	12.8	15.9	18.6	11.28	16.4
Data25	7.965	9.458	17.69	13.47	28.6	24.6	31.2	22.896
Data26	2.58	4.967	1.987	3.682	6.74	8.519	7.359	6.531
Data27	2139	2763	2516	3009	3896	4205	5009	4698
Data28	754.9	983.6	1250.96	1009.4	1936.4	2156.8	1769.39	1957.9
Data29	3.67	5.29	4.75	7.69	9.37	11.24	13.82	9.82
Data30	5.69	8.17	8.33	7.85	12.66	14.95	15.81	13.47

According to Table 6, the performance of S-shaped function-based methods has outperformed other methods in determining important features. The GTOAS1-based method yielded much more effective results than other S-shaped function-based methods. BGTOAS2 and BGTOAS3 also had an acceptable performance among the 30 datasets. The BGTOAS4 yielded the poorest results. The BGTOAV1 algorithm had the best results among the V-shaped function-based methods. BGTOAV2, BGTOAV3, and BGTOAV4 also produced poor results for the average number of selected features. Afterward, we analyzed the S-shaped and V-shaped algorithm results for the average objective function, presented in Table 7.

Table 7: Comparing the Proposed BGTOA Methods based on S-shaped and V-shaped Functions for the criterion of Average Objective Function Value

Datasets	BGTOAS1	BGTOAS2	BGTOAS3	BGTOAS4	BGTOAV1	BGTOAV2	BGTOAV3	BGTOAV4
Data1	0.091	0.089	30.06	0.109	40.103	70.036	30.085	**30.158**
Data2	0.016	0.024	0.04	0.032	50.289	40.045	60.211	**50.344**
Data3	0.297	0.301	0.26	10.264	20.277	50.183	80.065	**60.138**
Data4	0.67	0.666	0.67	0.665	0.726	40.437	70.233	**80.135**
Data5	0.478	0.512	0.5	0.507	50.239	80.095	40.298	**100**
Data6	10.298	0.332	0.33	10.298	20.264	70.099	50.164	**40.198**
Data7	10.53	0.568	0.614	0.596	50.36	30.476	50.337	**10.557**
Data8	0.066	0.171	0.073	0.068	70.096	70.224	50.376	**50.394**
Data9	0.058	0.046	0.103	0.07	60.202	80.123	80.07	**100**
Data10	0.02	10.033	0.041	20.019	60.057	50.038	80.028	**30.087**
Data11	0.148	0.181	0.065	0.106	20.182	20.312	70.097	**60.135**
Data12	30.292	0.439	0.455	0.407	80.071	70.099	50.219	**50.226**
Data13	0.371	0.428	0.457	0.355	70.13	80.086	20.346	**60.173**
Data14	0.393	0.392	0.392	0.393	40.233	70.145	20.325	**40.27**
Data15	0.274	0.222	0.171	0.117	80.058	70.09	50.228	**30.318**
Data16	50.21	10.377	30.24	0.346	70.14	60.165	40.217	**60.165**
Data17	0.333	0.334	0.331	0.332	30.231	60.132	60.132	**50.165**
Data18	0.25	40.151	0.252	0.252	60.099	70.074	90.025	**30.173**
Data19	0.126	0.093	0.136	0.071	40.212	30.33	70.115	**70.112**
Data20	0.478	0.486	0.476	0.476	60.202	60.198	10.458	**60.198**
Data21	0.039	0.04	0.039	0.039	80.006	80.055	60.105	**60.141**
Data22	0.065	0.053	0.059	0.054	40.043	40.051	100	**80.018**
Data23	0.61	0.61	0.611	0.61	20.511	40.469	50.388	**10.631**
Data24	0.029	0.045	0.026	0.033	100	20.032	20.038	**80.018**
Data25	0.591	0.595	0.555	0.56	50.401	90.057	80.176	**70.239**
Data26	20.27	0.346	10.31	10.31	100	80.06	40.207	**40.207**
Data27	0.298	0.296	0.288	0.293	40.183	70.086	40.169	**50.15**
Data28	0.614	0.616	0.614	0.617	60.253	60.256	50.323	**70.195**
Data29	0.444	0.461	0.36	0.485	60.157	40.254	70.154	**60.222**
Data30	40.343	0.567	0.566	0.565	60.272	60.226	50.344	**40.354**

According to Table 7, S-shaped function-based methods have outperformed other methods in determining objective function values. The BGTOAS1 yielded much more effective results than other S-shaped methods. Also, BGTOAS2 and BGTOAS4 methods had an acceptable performance among the 30 datasets. BGTOAS3 has the weakest results among the other S-shaped methods. BGTOAV1 and BGTOAV2 algorithms had the strongest results among the V-shaped methods. BGTOAV3 and BGTOAV4 also yielded poor results for the average number of selected features. These results prove the superiority of the BGTOAS1 method. The third test of this section analyzes S-shaped and V-shaped-based algorithms based on the average accuracy. These results are reported in Table 8.

Table 8: Comparing the S-shaped and V-shaped-based Proposed BGTOA of Average Accuracy

Datasets	BGTOAS1	BGTOAS2	BGTOAS3	BGTOAS4	BGTOAV1	BGTOAV2	BGTOAV3	BGTOAV4
Data1	0.914	0.915	0.646	0.895	0.497	0.264	0.615	**0.542**
Data2	0.989	0.983	0.965	0.973	0.208	0.557	0.187	**0.152**
Data3	0.709	0.703	0.744	0.638	0.522	0.315	0.134	**0.261**
Data4	0.327	0.329	0.325	0.332	0.267	0.159	0.065	**0.064**
Data5	0.519	0.488	0.5	0.49	0.259	0.104	0.299	**0.113**
Data6	0.602	0.668	0.668	0.602	0.535	0.201	0.334	**0.401**
Data7	0.372	0.434	0.387	0.403	0.138	0.221	0.161	**0.34**
Data8	0.939	0.834	0.932	0.938	0.204	0.074	0.12	**0.102**

Data9	0.946	0.962	0.903	0.935	0.196	0.076	0.129	**0.951**
Data10	0.987	0.874	0.964	0.789	0.343	0.463	0.172	**0.613**
Data11	0.858	0.823	0.94	0.9	0.619	0.486	0.203	**0.264**
Data12	0.412	0.563	0.546	0.598	0.129	0.2	0.28	**0.275**
Data13	0.628	0.571	0.547	0.644	0.169	0.113	0.451	**0.226**
Data14	0.607	0.606	0.607	0.607	0.364	0.153	0.472	**0.327**
Data15	0.728	0.781	0.835	0.888	0.142	0.21	0.27	**0.379**
Data16	0.291	0.521	0.459	0.656	0.158	0.233	0.381	**0.234**
Data17	0.667	0.667	0.667	0.667	0.467	0.267	0.267	**0.333**
Data18	0.75	0.45	0.75	0.75	0.3	0.225	0.075	**0.525**
Data19	0.878	0.911	0.869	0.932	0.386	0.368	0.184	**0.188**
Data20	0.521	0.51	0.521	0.521	0.196	0.2	0.438	**0.2**
Data21	0.965	0.963	0.965	0.966	0.194	0.144	0.294	**0.258**
Data22	0.94	0.951	0.946	0.95	0.557	0.549	0.896	**0.182**
Data23	0.385	0.386	0.386	0.386	0.284	0.126	0.108	**0.263**
Data24	0.981	0.96	0.982	0.972	0.92	0.769	0.762	**0.182**
Data25	0.41	0.405	0.442	0.44	0.12	0.143	0.123	**0.158**
Data26	0.523	0.653	0.588	0.588	0.458	0.131	0.392	**0.392**
Data27	0.699	0.708	0.709	0.708	0.415	0.213	0.429	**0.349**
Data28	0.38	0.381	0.381	0.381	0.145	0.141	0.173	**0.103**
Data29	0.555	0.538	0.638	0.512	0.242	0.345	0.145	**0.176**
Data30	0.259	0.431	0.431	0.431	0.125	0.173	0.153	**0.243**

According to Table 8, the S-shaped-based methods have outperformed other methods in obtaining higher average accuracies. BGTOAS1 and BGTOAS4 yielded much more effective results than other S-shaped methods. Also, BGTOAS2 and BGTOAS3 methods had an acceptable performance among the 30 datasets. BGTOAV1 and BGTOAV4 algorithms yielded the most robust results among the V-shaped methods. BGTOAV2 and BGTOAV3 also produced poor results for the average number of selected features. These results demonstrate the superiority of BGTOAS1.

5.3. Analyzing the BGTOALC Method

In the current section, the BGTOALC method is compared with BMNABC, BPSO, BGSA, BCCSA, BGWO, BFPA, and BBA in terms of average selected features, average accuracy, average objective function value, convergence level of the objective function, and standard deviation of the objective function. The comparison results are presented in the figures and tables of this section. The parameters of each comparative algorithm are presented in Table 4. The first test of this section compares the number of selected features in the BGTOALC method with other metaheuristic methods such as BPSO, BGSA, BCCSA, BGWO, BFPA, BBA, and BMNABC, the results of which are presented in Table 9.

Table 9: Comparing the Number of Selected Features in the BGTOALC Method with Other Binary Metaheuristic Algorithms

Data	BGSA	BBA	BPSO	BFPA	BGWO	BMNABC	BCCSA	BGTOAS1	BGTOALC
Data1	**6**	6.3	7.496	10.8	8.3	6.4	9	12.3	**6**
Data2	13	9.6	15.483	22.3	15.1	7	12.45	**5.75**	10
Data3	**4**	6.55	8.867	11.65	12	6	6	7.32	5
Data4	236.85	351.25	480.613	655.4	467.95	326	242.55	423.2	**40.65**
Data5	8821.9	6919.6	9909.96	12849.75	10063.65	2276	5247	7852	**300.95**
Data6	**1**	11.25	12.425	21.05	8.15	**1**	3	2.0498	**1**
Data7	5	3.25	3.712	5.3	5.25	5	5	**1.2145**	5
Data8	**6**	29.1	31.08	44.45	37.4	13	19.5	15.214	**6**

Data9	**10**	7.45	15.488	19.1	21.15	10.1	11	12.302	11
Data10	3	3.85	4.75	6.15	4.35	3.6	7	2.025	**2**
Data11	14	**6.5**	12.149	15.25	18.95	9.9	18	7.214	13
Data12	**1**	3.45	3.726	6.45	3.3	2	1.15	3.214	2
Data13	**2**	7.35	6.994	11.7	7.7	3	4	5.324	5
Data14	171.05	218.3	292.326	395.1	408.8	117.7	166.2	214.31	**108**
Data15	15	17.2	19.342	23.2	21.95	19.95	18	9.3245	**8.63**
Data16	**2**	24.85	37.765	55.75	32.6	**2**	15	22.39	**2**
Data17	1199.25	1254.2	1608.449	2126.15	1634	**170.35**	847.15	1102	526
Data18	3	20.4	21.369	34.85	20.15	4	10.6	14.325	**2.15**
Data19	10	8.65	8.383	11.3	11.05	8	11.95	8.458	**7.55**
Data20	1440.25	1533.45	1922.138	2585.2	1953.95	1290.9	1032	1402.7	**1041**
Data21	257.7	177.4	252.782	321.65	381.4	151	202	96.2	**17.2**
Data22	342.05	298.45	375.716	488.1	563.05	322.25	219.4	247.3	**205**
Data23	222.15	348.4	486.42	661.15	470.8	**5.5**	246	395.1	301
Data24	3	4.75	8.246	12.35	6.1	2.95	3	2.36	**2.15**
Data25	3	13.6	17.436	22.6	17.35	3	8.7	7.965	**2**
Data26	**1**	4.4	2.932	5.7	1.85	**1**	**1**	2.58	**1**
Data27	2170.55	2079.05	2788.212	3700.4	2803.2	2512.65	1887	2139	**1791**
Data28	790.25	829.75	1165.33	1570.1	1140.6	**280.2**	600	754.9	321
Data29	**2**	5.65	4.348	8.45	3.15	**2**	**2**	3.67	**2**
Data30	**4**	5.3	8.062	10.95	9.1	5	**4**	5.69	**4**

The comparison results of the BGTOALC with other binary meta-heuristic algorithms for the number of selected features are shown in Table 9, indicating that the proposed BGTOAS1 method exhibited better performance in the first two datasets and acceptable performance in the other ones. In addition, the proposed BGTOALC method had a better average in 67% of the datasets. A closer look at the results shows that the proposed method in data sets of Data3, Data9, Data12, Data13, and Data28 has performed close to the best algorithm (i.e., about 17%). It should also be noted that in the proposed approach, the value weight of the number of selected features was about 0.01%, so it is expected that it focused more on accuracy. This can be seen in the 11th data set, which has reached 90% accuracy with 13 features, while the BBA algorithm achieved 76% accuracy with an average number of 6.5 features. As a result, to evaluate the proposed BGTOALC method better and fairer, we should examine the average accuracy as well as the average number of selected features. In the following, the BGTOALC method is compared with binary meta-heuristic algorithms based on average accuracy, as depicted in Table (10)

Table 10: Comparing the Average Accuracy of the BGTOALC Method with Other Binary Metaheuristic Algorithms

Data	BGSA	BBA	BPSO	BFPA	BGWO	BMNABC	BCCSA	BGTOAS1	BGTOALC
Data1	0.977	0.915	0.981	0.937	0.987	0.991	0.977	0.895	**0.991**
Data2	0.995	0.905	0.998	0.98	0.993	0.995	0.989	0.979	**1**
Data3	0. 989	0.972	0.981	0.977	0.989	0.99	0.986	0.968	**0.99**
Data4	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	**0.509**
Data5	0.532	0.523	0.531	0.523	0.532	0.532	0.532	0.529	**0.567**
Data6	0.648	0.648	0.648	0.648	0.648	0.648	0.648	0.486	**0.648**
Data7	0.484	0.422	0.484	0.467	0.473	0.484	0.484	0.267	**0.494**
Data8	0.680	0.423	0.687	0.439	0.588	0.68	0.535	0.446	**0.69**
Data9	0.982	0.8	0.974	0.906	0.968	0.982	0.947	0.893	**0.986**
Data10	0.991	0.928	0.979	0.969	0.971	0.989	0.977	0.878	**0.991**
Data11	0.884	0.764	0.883	0.833	0.875	0.891	0.866	0.819	**0.90**

Data12	0.617	0.571	0.617	0.607	0.613	0.617	0.617	0.584	**0.617**
Data13	0.705	0.593	0.662	0.596	0.667	0.705	0.679	0.617	**0.718**
Data14	0.959	0.94	0.956	0.943	0.951	**0.962**	0.95	0.946	0.958
Data15	0.946	0.663	0.917	0.777	0.921	0.937	0.852	0.767	**0.946**
Data16	0.283	0.283	0.283	0.283	0.283	0.283	0.283	0.278	**0.283**
Data17	0.822	0.813	0.822	0.807	0.821	0.824	0.824	0.819	**0.824**
Data18	1	0.969	1	0.975	1	1	1	1	**1**
Data19	**0.973**	0.809	0.93	0.901	0.946	0.959	0.932	0.863	0.959
Data20	0.479	0.479	0.479	0.479	0.479	0.479	0.479	0.479	**0.479**
Data21	0.698	0.618	0.684	0.629	0.703	0.665	0.658	0.642	**0.734**
Data22	0.857	0.773	0.835	0.782	0.831	0.855	0.823	0.775	**0.877**
Data23	0.764	0.764	0.764	0.764	0.764	0.764	0.764	0.643	**0.764**
Data24	0.998	0.957	0.998	0.998	0.999	0.998	0.998	0.896	**0.999**
Data25	0.172	0.096	0.172	0.115	0.172	0.182	0.172	0.144	**0.182**
Data26	0.666	0.599	0.666	0.666	0.666	0.666	0.666	0.466	**0.666**
Data27	0.686	0.686	0.686	0.686	0.686	**0.709**	0.686	0.707	0.686
Data28	1	1	1	1	1	1	1	1	**1**
Data29	0.73	0.711	0.73	0.717	0.727	0.71	0.71	0.644	**0.73**
Data30	1	0.921	0.996	0.972	0.997	1	0.98	0.938	**1**

The comparison results of the BGTOALC with other binary meta-heuristic algorithms for the average accuracy are shown in Table 10, suggesting that the BGTOALC method had higher average accuracy in 93% of the datasets. It performed similarly to the best algorithm in 7% of cases, too. The results prove that the proposed BGTOALC method, with a greater focus on accuracy, has prevented the lower number of selected features. With a closer look, it can be seen that the average accuracy of the proposed method in the data set with high dimensions such as Data4, Data5, Data17, Data21, Data22, Data23, and Data28 has achieved better accuracy as well as less number of features. The proposed BGTOAS1, however, showed much poorer performance, which proves that this method can be considered as a primary feature selection method. Thus, when high dimensions and higher accuracy are taken into account, the proposed BGTOALC method can be considered as a proper and reliable method. Tables 9 and 10 proved that the proposed BGTOALC outperformed the BMNABC, BPSO, BGSA, BCCSA, BGWO, BFPA, and BBA algorithms while performing better than the proposed BGTOAS1. Figures 7 to 11 depict the convergence level of the BGTOALC method compared with other algorithms to further prove our proposed method.

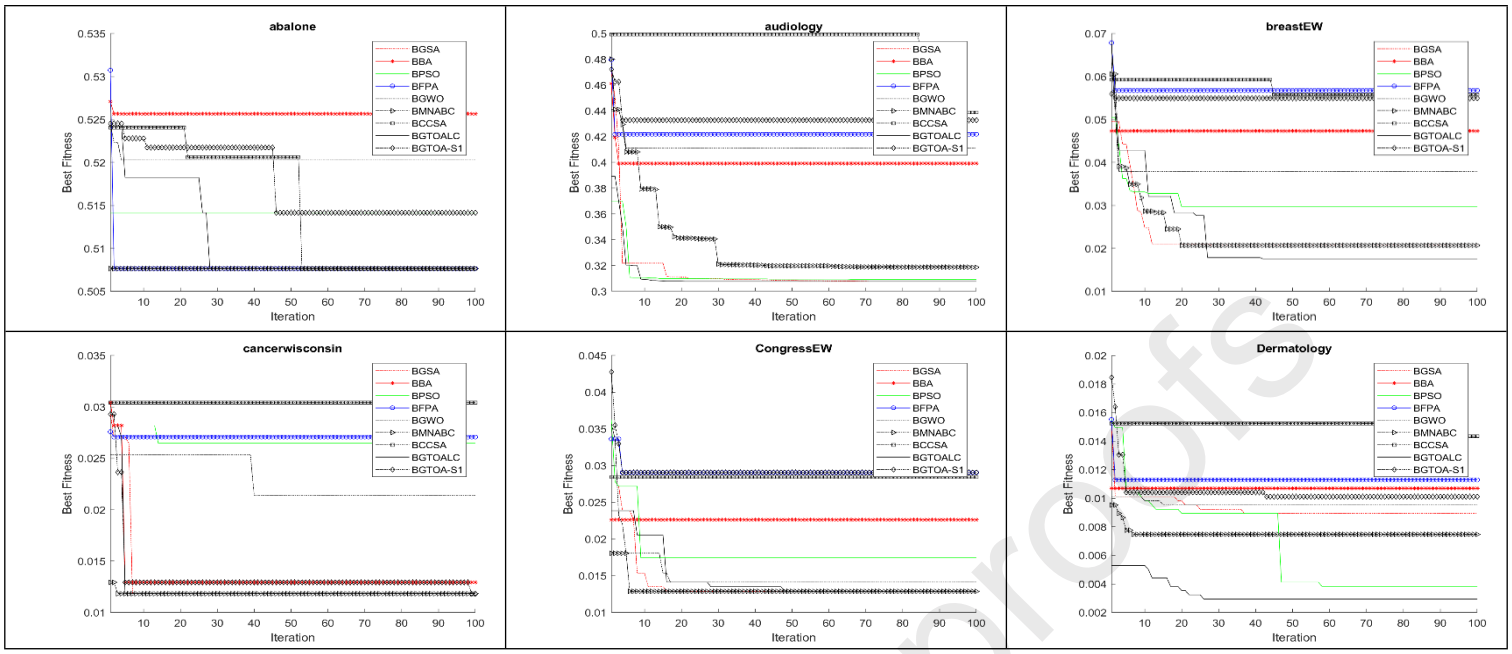


Figure 7: Comparing the Convergence Level of the BGTOALC in the first 6 sets

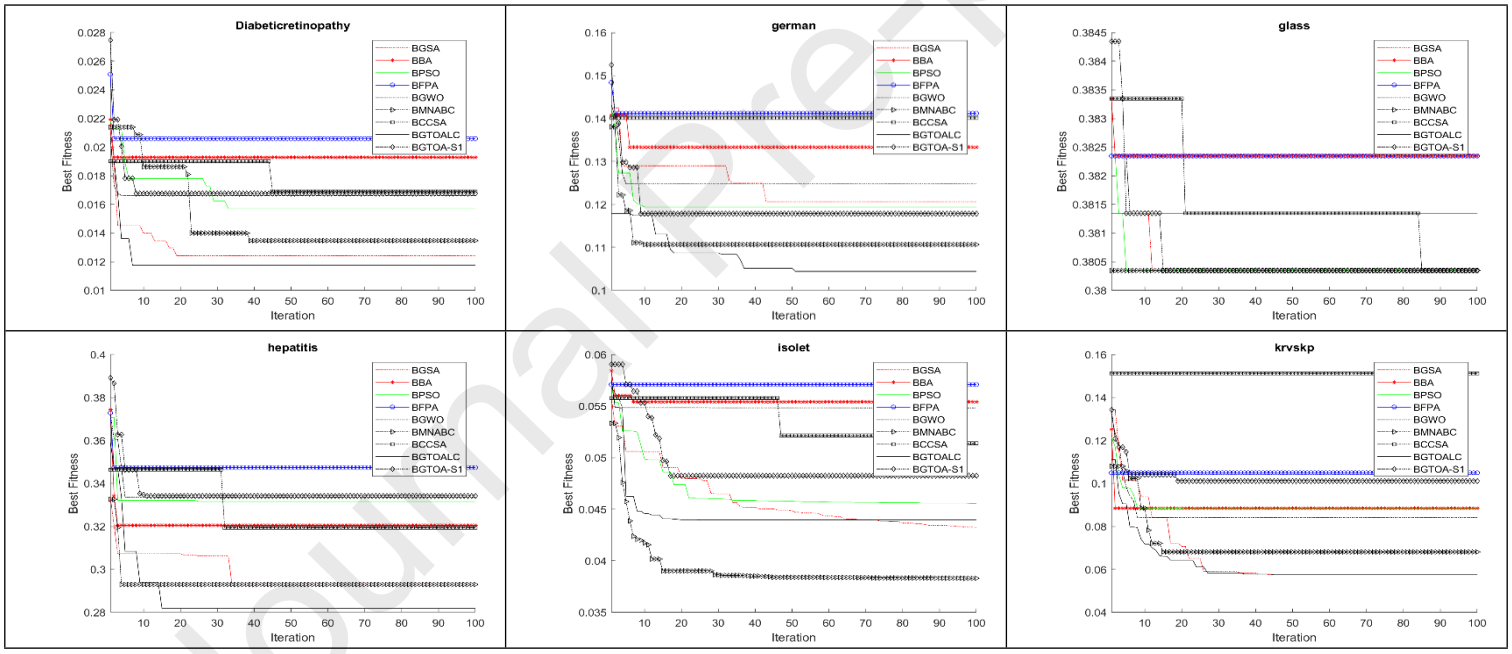


Figure 8: Comparing the Convergence Level of the BGTOALC in the second 6 sets

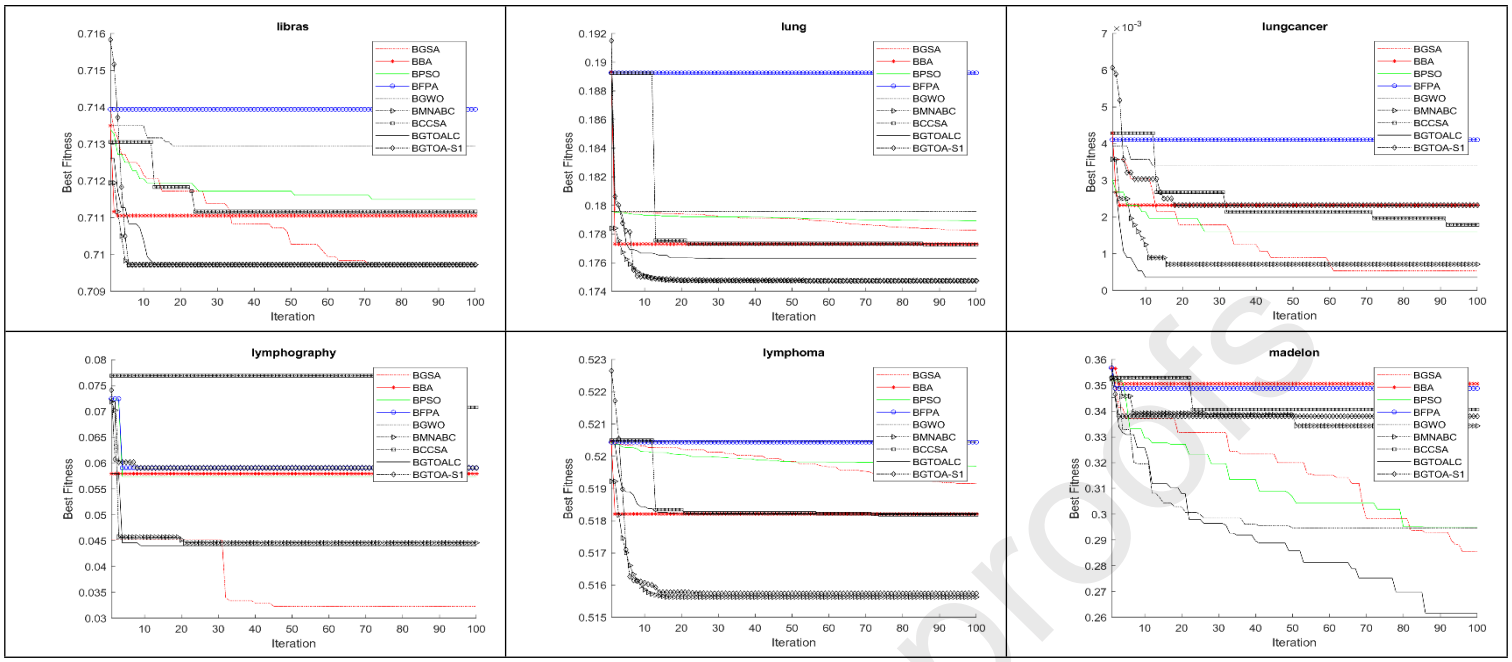


Figure 9: Comparing the Convergence Level of the BGTOALC in the third 6 sets

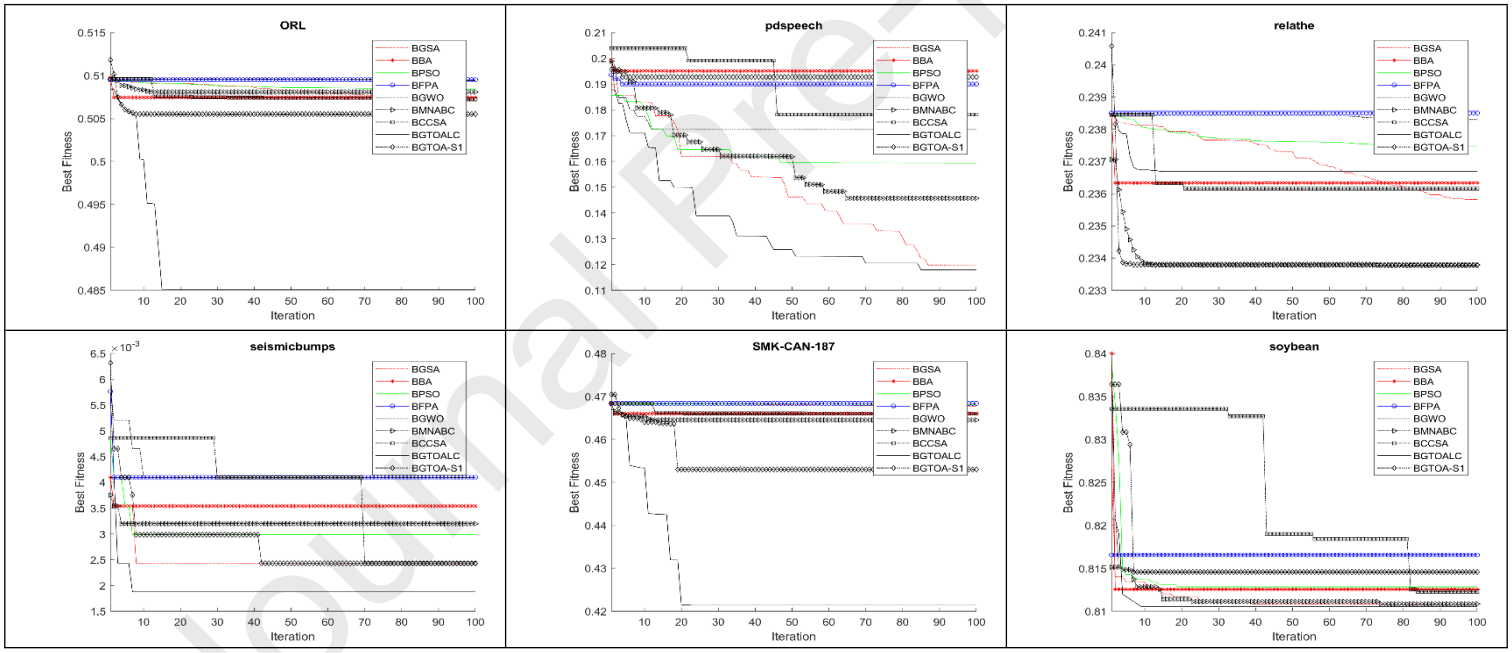


Figure 10: Comparing the Convergence Level of the BGTOALC in the fourth 6 sets

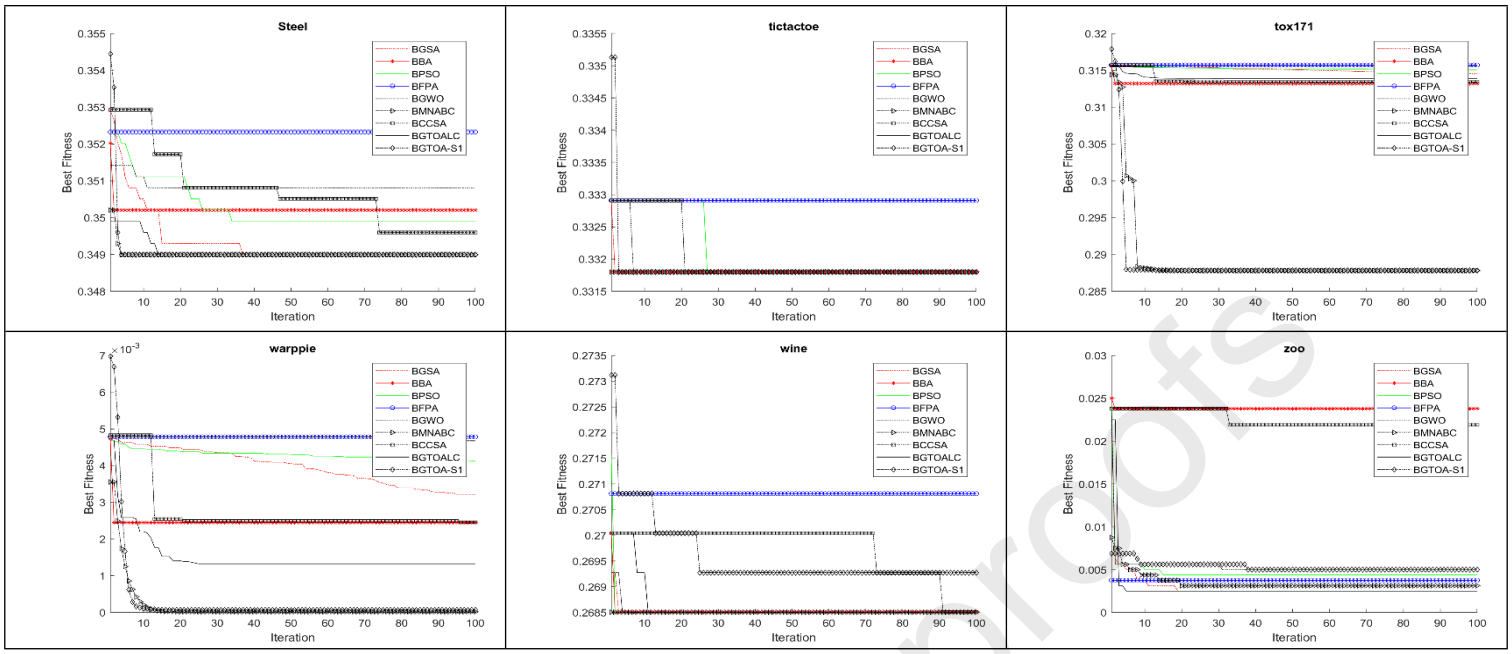
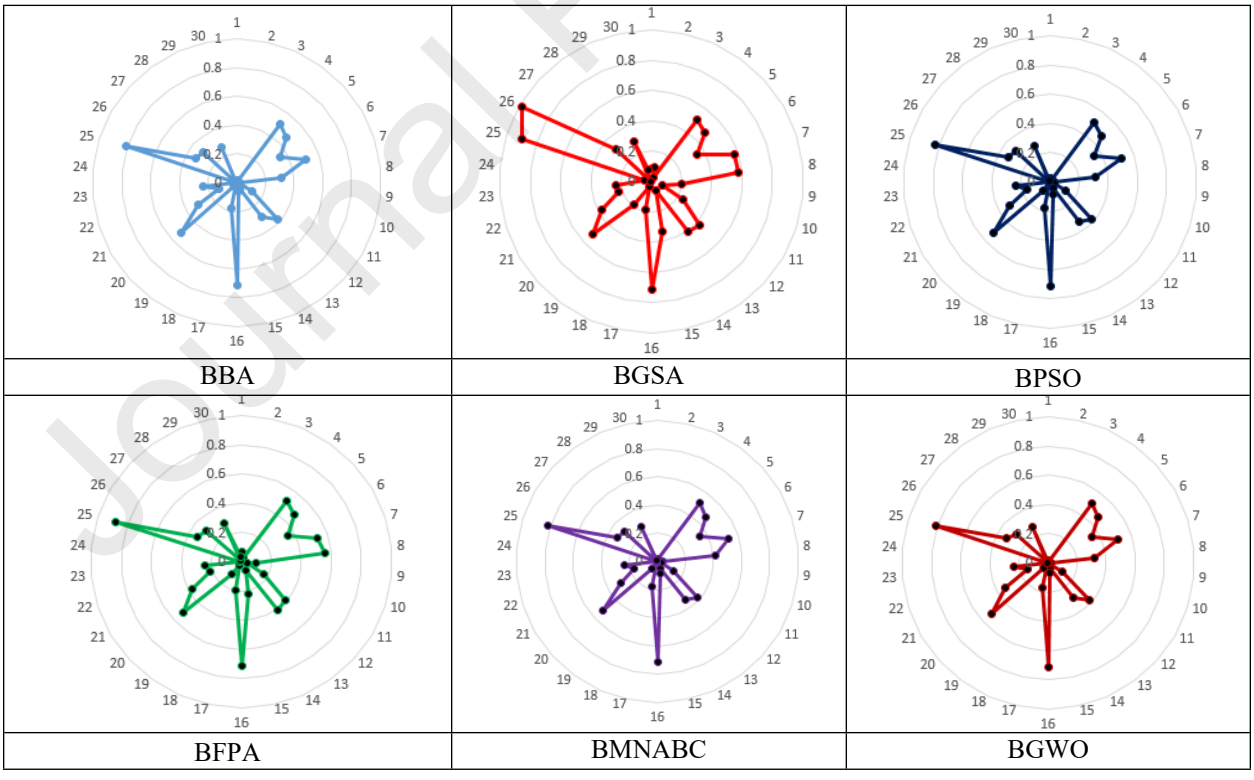


Figure 11: Comparing the Convergence Level of the BGTOALC in the fifth 6 sets

Figures 7 to 11 display the convergence level of the BGTOALC method compared with other algorithms. The proposed method usually had a higher convergence level in most datasets, while the BGTOAS1 dataset proved better in some other datasets. These convergence level results suggest that the BGTOALC method outperformed the BMNABC, BPSO, BGSA, BCCSA, BGWO, BFPA, and BBA algorithms. The average objective function and standard deviation are analyzed next to further prove the BGTOALC method.



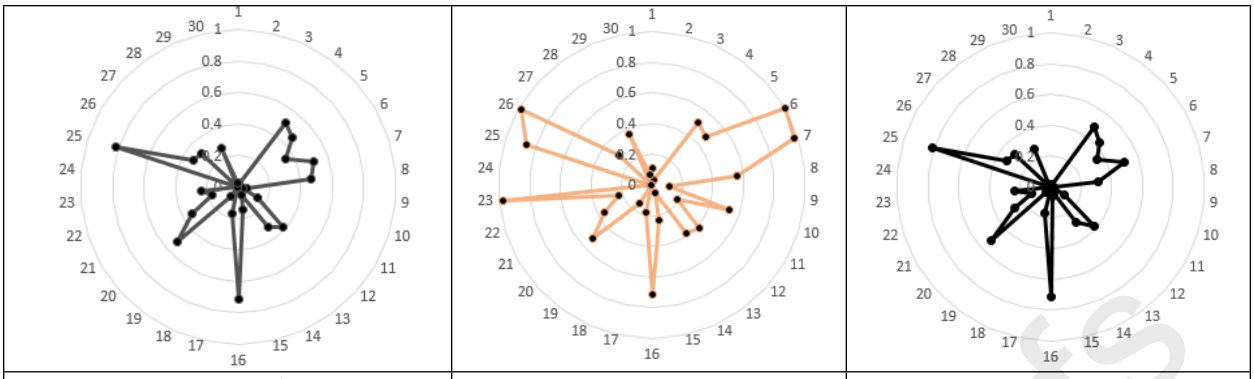


Figure 12: Comparing the Average Objective Function Values of the BGTOALC and Other Algorithms

Comparing the proposed BGTOALC method with other algorithms based on their average objective function values in Figure 12 shows the better performance of the proposed BGTOALC and BGSA algorithms regarding the centers. BGTOAS1 showed a generally poor performance in a manner that only the average objective function value of one dataset was close to 1and larger than 0.5. However, the proposed BGTOALC method reaches a value smaller than 0.5 in 90% of the datasets, which is rather promising in comparison with other algorithms. Afterward, the P-values of the BGTOALC method are compared with other based binary metaheuristic algorithms using the Wilcoxon rank test. The results of this test are reported in Table 11.

Table 11: Comparing the BGTOLAC Method p-values with Other Metaheuristic Algorithms Using the Wilcoxon rank test.

Data	BGSA	BBA	BPSO	BFPA	BGWO	BMNABC	BCCSA	BGTOAS1	BGTOALC
Data1	7.74E-06	8.86E-05	3.69E-05	8.83E-05	7.78E-05	2.93E-05	7.74E-06	7.74E-06	7.74E-06
Data2	7.74E-06	8.83E-05	7.58E-05	8.79E-05	7.53E-05	7.12E-05	2.30E-05	7.74E-06	7.74E-06
Data3	7.74E-06	8.79E-05	8.58E-05	8.83E-05	7.71E-05	7.93E-05	7.74E-06	5.40E-05	7.74E-06
Data4	8.68E-05	8.82E-05	8.57E-05	8.82E-05	8.44E-05	7.74E-06	5.30E-05	6.91E-05	7.74E-06
Data5	8.83E-05	8.86E-05	8.76E-05	8.86E-05	8.83E-05	8.61E-05	7.74E-06	8.86E-05	7.74E-06
Data6	7.74E-06	8.38E-05	5.06E-05	8.39E-05	6.21E-05	6.72E-05	7.74E-06	7.74E-06	7.74E-06
Data7	7.74E-06	8.84E-05	1.19E-05	8.76E-05	8.65E-05	7.69E-05	7.74E-06	7.74E-06	7.74E-06
Data8	7.74E-06	8.86E-05	4.37E-05	8.86E-05	6.66E-05	6.85E-05	5.40E-05	7.74E-06	7.74E-06
Data9	7.74E-06	8.84E-05	2.97E-05	8.84E-05	4.94E-05	8.13E-05	7.74E-06	7.74E-06	1.71E-05
Data10	7.74E-06	8.78E-05	2.96E-05	8.83E-05	8.11E-05	2.97E-05	7.74E-06	7.74E-06	2.93E-05
Data11	7.74E-06	8.84E-05	3.68E-05	8.86E-05	8.43E-05	5.88E-05	7.74E-06	7.74E-06	4.15E-05
Data12	7.74E-06	8.46E-05	7.74E-06	8.72E-05	8.29E-05	4.94E-05	2.30E-05	7.74E-06	7.74E-06
Data13	7.74E-06	8.54E-05	2.97E-05	8.66E-05	6.12E-05	7.68E-05	7.74E-06	7.74E-06	7.74E-06
Data14	8.73E-05	8.86E-05	8.70E-05	8.84E-05	7.48E-05	5.60E-05	1.71E-05	7.74E-06	8.66E-05
Data15	7.74E-06	8.86E-05	2.31E-05	8.86E-05	7.22E-05	6.20E-05	7.74E-06	5.31E-05	7.74E-06
Data16	7.74E-06	8.72E-05	5.57E-05	8.61E-05	7.11E-05	2.30E-05	7.74E-06	7.74E-06	7.74E-06
Data17	8.82E-05	8.86E-05	8.81E-05	8.86E-05	8.16E-05	3.66E-05	1.19E-05	7.74E-06	8.63E-05
Data18	7.74E-06	8.65E-05	2.93E-05	8.75E-05	6.21E-05	1.71E-05	5.06E-05	2.30E-05	7.74E-06
Data19	7.74E-06	8.83E-05	5.08E-05	8.77E-05	6.12E-05	6.82E-05	5.31E-05	5.62E-05	7.74E-06
Data20	8.81E-05	8.84E-05	8.72E-05	8.84E-05	8.68E-05	3.67E-05	7.74E-06	7.74E-06	7.59E-05
Data21	8.86E-05	8.86E-05	8.84E-05	8.86E-05	8.35E-05	1.19E-05	7.74E-06	4.40E-05	7.74E-06
Data22	8.86E-05	8.86E-05	8.83E-05	8.86E-05	8.77E-05	7.59E-05	7.74E-06	7.22E-05	1.71E-05
Data23	8.77E-05	8.84E-05	8.14E-05	8.81E-05	8.40E-05	8.84E-05	7.74E-06	7.74E-06	7.23E-05
Data24	7.74E-06	8.82E-05	2.31E-05	8.75E-05	5.57E-05	6.95E-05	7.74E-06	2.30E-05	7.74E-06
Data25	7.74E-06	8.81E-05	6.07E-05	8.81E-05	7.39E-05	2.97E-05	4.15E-05	7.74E-06	7.74E-06
Data26	7.74E-06	7.94E-05	2.93E-05	7.88E-05	6.03E-05	7.35E-05	7.74E-06	7.74E-06	7.74E-06
Data27	8.86E-05	8.86E-05	8.75E-05	8.84E-05	8.78E-05	3.68E-05	7.74E-06	7.74E-06	8.77E-05
Data28	8.76E-05	8.84E-05	8.67E-05	8.83E-05	8.60E-05	7.44E-05	5.06E-05	7.74E-06	8.56E-05

Data29	7.74E-06	8.71E-05	2.30E-05	8.72E-05	8.20E-05	5.61E-05	7.74E-06	7.74E-06	7.74E-06
Data30	7.74E-06	8.75E-05	6.20E-05	8.70E-05	8.19E-05	7.39E-05	7.74E-06	7.74E-06	7.74E-06

Comparing the P-values of the proposed BGTOALC method with other meta-heuristic algorithms using the Wilcoxon rank test in Table 11 shows a P-value lower than 0.05 for the proposed BGTOALC method in all the datasets. In addition, the proposed BGTOALC method has appeared in the first and second datasets, such as the powerful BGSA and BCCSA algorithms. Although the proposed BGTOALC method has higher p-values in some datasets like ninth, tenth, and twenty-third than methods such as BGSA and BGTOAS1, the overall results prove that the proposed BGTOALC method can be a reliable method for feature selection in different dimensions.

5.4. Comparing the BGTOALC Method with Filter-based Methods

In this section, the BGTOALC method is compared with different filter-based methods. In these methods, each feature is given a score, and the number of important features is not known. Therefore, the number of features has been set to 5 for low-dimensional and 10 for high-dimensional datasets. The following filter-based methods are used in this section:

- Fisher Score (F-Score)
- Univariate feature ranking for regression using F-tests(F-tests)
- Rank importance of predictors using ReliefF or RReliefF algorithm(ReliefF)
- Univariate feature ranking for classification using chi-square tests(chi-square)
- Rank features for classification using minimum redundancy maximum relevance (MRMR)

The BGTOALC method accuracy comparison results with the F-Score, F-Tests, ReliefF, and Chi-Square filter-based methods are presented in Table 12.

Table 12: Comparing the BGTOALC method accuracy with filter-based methods over 30 datasets

Data	F-Score	F-tests	ReliefF	chi-square	MRMR	BGTOALC
Data1	0.9309	0.9309	0.9309	0.9309	0.9309	**0.9677**
Data2	0.8087	0.8087	0.8087	0.8087	0.8087	**1**
Data3	0.9497	0.9497	0.9497	0.9497	0.9497	**0.99**
Data4	0.57	0.57	0.57	0.57	0.57	**0.57**
Data5	0.523	0.523	0.529	0.529	0.529	**0.567**
Data6	0.6715	0.6715	0.6715	0.6715	0.6715	**0.6715**
Data7	0.2379	0.2379	0.2379	0.2379	0.2379	**0.494**
Data8	0.66	0.66	0.66	0.66	0.66	**0.77**
Data9	0.9474	0.9474	0.9474	0.9474	0.9474	**0.986**
Data10	0.9514	0.9514	0.9514	0.9514	0.9514	**0.9943**
Data11	0.9	0.9	0.9	0.9	0.9	**0.942**
Data12	0.5327	0.5327	0.5327	0.5327	0.5327	**0.6262**
Data13	0.9487	0.9487	0.9487	0.9487	0.9487	**0.9487**
Data14	0.8397	0.8397	0.8397	0.8397	0.8397	**0.958**
Data15	0.6514	0.6514	0.6514	0.6514	0.6514	**0.9781**
Data16	0.2111	0.2111	0.2111	0.2111	0.2111	**0.283**
Data17	0.9902	0.9902	0.9902	0.9902	0.9902	**1**
Data18	0.9375	0.9375	0.9375	0.9375	0.9375	**1**
Data19	0.8243	0.8243	0.8243	0.8243	0.8243	**0.973**
Data20	0.6667	0.6667	0.6667	0.6667	0.6667	**0.6667**
Data21	0.6423	0.6423	0.6423	0.6423	0.6423	**0.7432**

Data22	0.9947	0.9947	0.9947	0.9947	0.9947	**0.9947**
Data23	0.175	0.175	0.175	0.175	0.175	**0.7764**
Data24	0.9567	0.9567	0.9567	0.9567	0.9567	**0.999**
Data25	0.474	0.474	0.474	0.474	0.474	0.2662
Data26	0.6764	0.6764	0.6764	0.6764	0.6764	**0.6764**
Data27	0.7326	0.7326	0.7326	0.7326	0.7326	**0.7326**
Data28	0.8952	0.8952	0.8952	0.8952	0.8952	**1**
Data29	0.7416	0.7416	0.7416	0.7416	0.7416	**0.7416**
Data30	0.902	0.902	0.902	0.902	0.902	**1**

The proposed BGTOALC method accuracy comparison with filter-based methods over 30 datasets shows that our proposed method has higher accuracy in 96% of the datasets compared to the F-Score, F-Tests, ReliefF, and Chi-Square filter-based methods. Of course, the accuracy results of the proposed BGTOALC are obtained independently through 20 iterations. The proposed BGTOALC method in some data sets, such as Data 7, Data 8, Data15, etc., is able to select more important features that have achieved higher accuracy compared to other methods. Since the Accuracy criterion may not take into account some other aspects, such as True Positive Rate and True Negative Rate, we will review and evaluate the classification criteria, including Precision, Sensitivity, and Specificity, too. The BGTOALC method precision comparison results with the F-Score, F-Tests, ReliefF, and Chi-Square filter-based methods are presented in Table 13.

Table 13: Comparing the BGTOALC Method Precision with Filter-based Methods over 30 Datasets

Data	F-Score	F-tests	ReliefF	chi-square	MRMR	BGTOALC
Data1	0.9585	0.9585	0.9585	0.9585	0.9585	**0.977**
Data2	0.9617	0.9617	0.9617	0.9617	0.9617	**0.9836**
Data3	0.9132	0.9132	0.9132	0.9132	0.9132	**0.9219**
Data4	0.33	0.33	0.33	0.33	0.33	**0.33**
Data5	0.8191	0.8191	0.8191	0.8191	0.8191	0.7979
Data6	0.6612	0.6612	0.6612	0.6612	0.6612	**0.6612**
Data7	0.2135	0.2135	0.2135	0.2135	0.2135	**0.2135**
Data8	0.86	0.86	0.86	0.86	0.86	0.77
Data9	0.6807	0.6807	0.6807	0.6807	0.6807	**0.8316**
Data10	0.9743	0.9743	0.9743	0.9743	0.9743	**0.9857**
Data11	0.748	0.748	0.748	0.748	0.748	**0.796**
Data12	0.7103	0.7103	0.7103	0.7103	0.7103	**0.7103**
Data13	0.5128	0.5128	0.5128	0.5128	0.5128	**0.5641**
Data14	0.309	0.309	0.309	0.309	0.309	**0.309**
Data15	0.719	0.719	0.719	0.719	0.719	**0.9518**
Data16	0.7222	0.7222	0.7222	0.7222	0.7222	**0.7222**
Data17	0.7353	0.7353	0.7353	0.7353	0.7353	**0.7353**
Data18	0.6875	0.6875	0.6875	0.6875	0.6875	**0.6875**
Data19	0.7838	0.7838	0.7838	0.7838	0.7838	**0.9054**
Data20	0.4583	0.4583	0.4583	0.4583	0.4583	**0.4583**
Data21	0.8123	0.8123	0.8123	0.8123	0.8123	**0.8154**
Data22	0.8413	0.8413	0.8413	0.8413	0.8413	**0.8492**
Data23	0.4431	0.4431	0.4431	0.4431	0.4431	**0.4361**
Data24	0.9683	0.9683	0.9683	0.9683	0.9683	**0.969**

Data25	0.9091	0.9091	0.9091	0.9091	0.9091	**0.9091**
Data26	0.6889	0.6889	0.6889	0.6889	0.6889	0.666
Data27	0.5698	0.5698	0.5698	0.5698	0.5698	**0.6047**
Data28	0.781	0.781	0.781	0.781	0.781	0.7714
Data29	0.6854	0.6854	0.6854	0.6854	0.6854	**0.6854**
Data30	0.9412	0.9412	0.9412	0.9412	0.9412	**0.9804**

The proposed BGTOALC method precision comparison with filter-based methods over 30 datasets shows that our proposed method has higher accuracy in 86% of the datasets compared to the F-Score, F-Tests, ReliefF, and Chi-Square filter-based methods. The proposed BGTOALC has been able to improve the accuracy and precision criteria. In addition, we further assessed the Sensitivity and Specificity criteria, which focus on the True Positive Rate and the True Negative Rate, respectively. The results of these two criteria are very important, along with the two criteria of accuracy and precision, and will prove the superiority of the BGTOALC. The BGTOALC method sensitivity comparison results with the F-Score, F-Tests, ReliefF, and Chi-Square filter-based methods can be seen in Table 14.

Table 14: Comparing the BGTOALC Method Sensitivity with Filter-based Methods over 30 Datasets

Data	F-Score	F-tests	ReliefF	chi-square	MRMR	BGTOALC
Data1	0.9585	0.9585	0.9585	0.9585	0.9585	**0.977**
Data2	0.9617	0.9617	0.9617	0.9617	0.9617	**0.9836**
Data3	0.9132	0.9132	0.9132	0.9132	0.9132	**0.9219**
Data4	0.33	0.33	0.33	0.33	0.33	**0.33**
Data5	0.8191	0.8191	0.8191	0.8191	0.8191	0.7979
Data6	0.6612	0.6612	0.6612	0.6612	0.6612	**0.6612**
Data7	0.2135	0.2135	0.2135	0.2135	0.2135	**0.2135**
Data8	0.86	0.86	0.86	0.86	0.86	0.77
Data9	0.6807	0.6807	0.6807	0.6807	0.6807	**0.8316**
Data10	0.9743	0.9743	0.9743	0.9743	0.9743	**0.9857**
Data11	0.748	0.748	0.748	0.748	0.748	**0.796**
Data12	0.7103	0.7103	0.7103	0.7103	0.7103	**0.7103**
Data13	0.5128	0.5128	0.5128	0.5128	0.5128	**0.5641**
Data14	0.309	0.309	0.309	0.309	0.309	**0.309**
Data15	0.719	0.719	0.719	0.719	0.719	**0.9518**
Data16	0.7222	0.7222	0.7222	0.7222	0.7222	**0.7222**
Data17	0.7353	0.7353	0.7353	0.7353	0.7353	**0.7353**
Data18	0.6875	0.6875	0.6875	0.6875	0.6875	**0.6875**
Data19	0.7838	0.7838	0.7838	0.7838	0.7838	**0.9054**
Data20	0.4583	0.4583	0.4583	0.4583	0.4583	**0.4583**
Data21	0.8123	0.8123	0.8123	0.8123	0.8123	**0.8154**
Data22	0.8413	0.8413	0.8413	0.8413	0.8413	**0.8492**
Data23	0.4431	0.4431	0.4431	0.4431	0.4431	**0.4361**
Data24	0.9683	0.9683	0.9683	0.9683	0.9683	**0.969**
Data25	0.9091	0.9091	0.9091	0.9091	0.9091	**0.9091**
Data26	0.6889	0.6889	0.6889	0.6889	0.6889	0.666
Data27	0.5698	0.5698	0.5698	0.5698	0.5698	**0.6047**
Data28	0.781	0.781	0.781	0.781	0.781	0.7714
Data29	0.6854	0.6854	0.6854	0.6854	0.6854	**0.6854**

Data30	0.9412	0.9412	0.9412	0.9412	0.9412	**0.9804**

The proposed BGTOALC method sensitivity comparison with filter-based methods over 30 datasets shows that our proposed method has higher accuracy in 86% of the datasets compared to the F-Score, F-Tests, ReliefF, and Chi-Square filter-based methods. The results of the test show that the proposed method has achieved significant superiority in terms of sensitivity. In addition, for final approval, we analyzed the Specificity criterion. The results of comparing the proposed BGTOALC method with methods based on F-Score, F-tests, ReliefF, and Chi-Square filter-based methods in terms of Sensitivity criteria are shown in Table 15.

Table 15: Comparing the BGTOALC Method Specificity with Filter-based Methods over 30 Datasets

Data	F-Score	F-tests	ReliefF	chi-square	MRMR	BGTOALC
Data1	0.9585	0.9585	0.9585	0.9585	0.9585	**0.977**
Data2	0.9923	0.9923	0.9923	0.9923	0.9923	**0.9967**
Data3	0.9132	0.9132	0.9132	0.9132	0.9132	**0.9219**
Data4	0.9828	0.9828	0.9828	0.9828	0.9828	**0.9828**
Data5	0.8191	0.8191	0.8191	0.8191	0.8191	0.7979
Data6	0.6612	0.6612	0.6612	0.6612	0.6612	**0.6612**
Data7	0.9685	0.9685	0.9685	0.9685	0.9685	**0.9685**
Data8	0.9922	0.9922	0.9922	0.9922	0.9922	**0.9879**
Data9	0.6807	0.6807	0.6807	0.6807	0.6807	**0.8316**
Data10	0.9743	0.9743	0.9743	0.9743	0.9743	**0.9857**
Data11	0.748	0.748	0.748	0.748	0.748	**0.796**
Data12	0.9421	0.9421	0.9421	0.9421	0.9421	**0.9421**
Data13	0.5128	0.5128	0.5128	0.5128	0.5128	**0.5641**
Data14	0.9724	0.9724	0.9724	0.9724	0.9724	**0.9724**
Data15	0.719	0.719	0.719	0.719	0.719	**0.9518**
Data16	0.9802	0.9802	0.9802	0.9802	0.9802	0.9794
Data17	0.9338	0.9338	0.9338	0.9338	0.9338	**0.9338**
Data18	0.6875	0.6875	0.6875	0.6875	0.6875	**0.6875**
Data19	0.8919	0.8919	0.8919	0.8919	0.8919	**0.9527**
Data20	0.9323	0.9323	0.9323	0.9323	0.9323	0.9097
Data21	0.8123	0.8123	0.8123	0.8123	0.8123	**0.8154**
Data22	0.8413	0.8413	0.8413	0.8413	0.8413	**0.8492**
Data23	0.9707	0.9707	0.9707	0.9707	0.9707	0.9703
Data24	0.9683	0.9683	0.9683	0.9683	0.9683	**0.969**
Data25	0.9947	0.9947	0.9947	0.9947	0.9947	**0.9947**
Data26	0.6889	0.6889	0.6889	0.6889	0.6889	**0.666**
Data27	0.8566	0.8566	0.8566	0.8566	0.8566	**0.8682**
Data28	0.9757	0.9757	0.9757	0.9757	0.9757	0.9746
Data29	0.8427	0.8427	0.8427	0.8427	0.8427	**0.8427**
Data30	0.9902	0.9902	0.9902	0.9902	0.9902	**0.9967**

The comparison of the proposed BGTOALC method with filter-based methods in terms of specificity on 30 datasets shows that the proposed method is significantly superior in approximately 83% of the datasets to the F-Score, F-tests, ReliefF, and Chi-square methods. The proposed BGTOALC method, also in several datasets, such as Data23, Data16, and Data28, is very close to the superior method that can be considered very promising results. All the test results obtained in this paper showed that the BGTOALC method with its entire novel

operators has turned into an accurate algorithm compared to other continuous, transfer, and binary metaheuristic algorithms. Moreover, these results show that the BGTOALC is capable of maintaining its performance in high dimensional conditions, which can turn it into a powerful method of solving binary problems.

6. Conclusions

This paper implemented three different binary versions of the GTOA in solving feature selection problems. The first two versions, BGTOAV and BGTOAS, were designed utilizing S-shaped and V-shaped transfer functions, while the third version, BGTOALC, was developed based on local search and chaos mapping, increasing the exploration and exploitation capabilities of the algorithm. The proposed BGTOALC method applied fundamental changes to the continuous GTOA using piecewise chaotic mapping to generate initial binary populations. This technique led to the generation of a higher quality initial population and increased the initial exploration of the method. There were also more enhancements in the teacher phase using the novel BTPGG and BTPGB operators, the student phase using the novel BSOBL operator, and the teacher allocation phase using the novel MBS operator, to maintain the balance between exploration and exploitation capabilities in solving feature selection problems. The proposed method was evaluated over 30 well-known datasets with different dimensions regarding various criteria, such as average accuracy, the average number of selected features, and average objective function value. The obtained results demonstrated that the BGTOAS1 method outperformed other S-shaped and V-shaped transfer functions based methods. Eventually, the BGTOALC method was compared with BGTOAS1 as the best-developed transfer function-based method and other previous binary metaheuristic algorithms, such as BBA, BMNABC, BPSO, BGSA, BCCSA, BGWO, and BFPA. The statistical analyses showed that the BGTOALC method had a better performance regarding an average number of features in 67% of datasets. Also, the obtained results were identical to the best algorithm in 23% of cases. In addition, it had better average accuracy in 93% of datasets while performing the same as the best algorithm in 7% of cases. Moreover, convergence rate analysis and the Wilcoxon rank test proved the considerable superiority of the BGTOALC over the other algorithms. Given the promising results of the proposed BGTOALC algorithm, in the future, this algorithm can be used for various areas like sentiment analysis, computer network intrusion detection, clustering, medical imaging, and scheduling problems with different objective functions and constraints. Also, the GTOA algorithm is structured in a way that the balance between exploration and exploitation is generally adjusted by population division. Therefore, one of the future goals is to provide a dynamic version of the algorithm to balance exploration and exploitation. Another major goal would be to develop a multi-objective version of the proposed algorithm so that we can have a Pareto optimality solution with the goal of less number of selected features and greater accuracy. Also, in this article, we have used the KNN algorithm as the classification algorithm, and in future studies, we will use other machine learning algorithms, including support vector algorithm (SVM), artificial neural network (NN), etc., to get different results. Besides, as mentioned earlier, there are numerous representative computational intelligence algorithms in the literature, which can be studied to address the feature selection problem in high-dimensional data, e.g., MBO (Wang, Deb & Cui, 2019), EWA (Wang, Deb & Coelho, 2018), EHO (Wang, Deb & Coelho, 2015), MS algorithm (Wang, 2018), SMA (Li, Chen, Wang, Heidari & Mirjalili, 2020), HGS (Yang, Chen, Heidari & Gandomi, 2021), CPA (Tu, Chen, Wang & Gandomi, 2021), HHO (Heidari et al., 2019), and FFA (Shayanfar & Gharehchopogh, 2018), in future works.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Acknowledgement

We thank Dr. Bagher Bayat (Research Scientist, Institute of Bio- and Geosciences IBG-3: Agrosphere) for his comments that greatly improved the manuscript.

References

Abd Elaziz, M., Ewees, A. A., Ibrahim, R. A., & Lu, S. (2020). Opposition-based moth-flame optimization improved by differential evolution for feature selection. Mathematics and Computers in 35Simulation, 168, 48-75. https://doi.org/10.1016/j.matcom.2019.06.017.

Abdel-Basset, M., El-Shahat, D., El-henawy, I., de Albuquerque, V. H. C., & Mirjalili, S. (2020). A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection. Expert Systems with Applications, 139, 112824. https://doi.org/10.1016/j.eswa.2019.112824

Abualigah, L. M. Q. (2019). Feature selection and enhanced krill herd algorithm for text document clustering. Springer (Chapter 1).

Alweshah, M., Al Khalaileh, S., Gupta, B. B., Almomani, A., Hammouri, A. I., & Al-Betar, M. A. (2020). The monarch butterfly optimization algorithm for solving feature selection problems. Neural Computing and Applications, 1-15. https://doi.org/10.1007/s00521-020-05210-0.

Arora, S., & Anand, P. (2019). Binary butterfly optimization approaches for feature selection. Expert Systems with Applications, 116, 147-160. https://doi.org/10.1016/j.eswa.2018.08.051.

Arora, S., & Singh, S. (2019). Butterfly optimization algorithm: a novel approach for global optimization. Soft Computing, 23(3), 715-734. https://doi.org/10.1007/s00500-018-3102-4.

Beheshti, Z. (2018). BMNABC: binary multi-neighborhood artificial bee colony for high-dimensional discrete optimization problems. Cybernetics and Systems, 49(7-8), 452-474. https://doi.org/10.1080/01969722.2018.1541597.

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. Computers & Electrical Engineering, 40(1), 16-28. https://doi.org/10.1016/j.compeleceng.2013.11.024.

Chaudhuri, A., & Sahu, T. P. (2021). Feature selection using Binary Crow Search Algorithm with time varying flight length. Expert Systems with Applications, 168, 114288. https://doi.org/10.1016/j.eswa.2020.114288.

De Souza, R. C. T., de Macedo, C. A., dos Santos Coelho, L., Pierezan, J., & Mariani, V. C. (2020). Binary coyote optimization algorithm for feature selection. Pattern Recognition, 107, 107470. https://doi.org/10.1016/j.patcog.2020.107470.

De Souza, R. C. T., dos Santos Coelho, L., De Macedo, C. A., & Pierezan, J. (2018). A V-shaped binary crow search algorithm for feature selection. Paper presented at the 2018 IEEE congress on evolutionary computation (CEC).

Ding, Y., Zhou, K., & Bi, W. (2020). Feature selection based on hybridization of genetic algorithm and competitive swarm optimizer. Soft Computing, 1-10. https://doi.org/10.1007/s00500-019-04628-6.

Emary, E., Zawbaa, H. M., & Hassanien, A. E. (2016a). Binary ant lion approaches for feature selection. Neurocomputing, 213, 54-65. https://doi.org/10.1016/j.neucom.2016.03.101.

Emary, E., Zawbaa, H. M., & Hassanien, A. E. (2016b). Binary grey wolf optimization approaches for feature selection. Neurocomputing, 172, 371-381. https://doi.org/10.1016/j.neucom.2015.06.083.

Emine, B., & Ülker, E. (2020). An efficient binary social spider algorithm for feature selection problem. Expert Systems with Applications, 146, 113185. https://doi.org/10.1016/j.eswa.2020.113185.

Faris, H., Mafarja, M. M., Heidari, A. A., Aljarah, I., Ala'M, A.-Z., Mirjalili, S., & Fujita, H. (2018). An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. Knowledge-Based Systems, 154, 43-67. https://doi.org/10.1016/j.knosys.2018.05.009.

Gonzalez-Lopez, J., Ventura, S., & Cano, A. (2020). Distributed multi-label feature selection using individual mutual information measures. Knowledge-Based Systems, 188, 105052. https://doi.org/10.1016/j.knosys.2019.105052.

Heidari, A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. Future Generation Computer Systems, 97, 849-872. https://doi.org/10.1016/j.future.2019.02.028.

Hu, P., Pan, J.-S., & Chu, S.-C. (2020). Improved binary grey wolf optimizer and its application for feature selection. Knowledge-Based Systems, 195, 105746. https://doi.org/10.1016/j.knosys.2020.105746.

Hussien, A. G., Hassanien, A. E., Houssein, E. H., Bhattacharyya, S., & Amin, M. (2019). S-shaped binary whale optimization algorithm for feature selection Recent trends in signal and image processing (pp. 79-87): Springer. https://doi.org/10.1007/978-981-10-8863-6_9.

Jiang, Y., Wu, Q., Zhang, G., Zhu, S., & Xing, W. (2021). A diversified group teaching optimization algorithm with segment-based fitness strategy for unmanned aerial vehicle route planning. Expert Systems with Applications, 185, 115690. https://doi.org/10.1016/j.eswa.2021.115690.

Khaire, U. M., & Dhanalakshmi, R. (2019). Stability of feature selection algorithm: A review. Journal of King Saud University-Computer and Information Sciences. https://doi.org/ 10.1016/j.jksuci.2019.06.012.

Khuat, T. T., & Le, M. H. (2019). Binary teaching–learning-based optimization algorithm with a new update mechanism for sample subset optimization in software defect prediction. Soft Computing, 23(20), 9919-9935. https://doi.org/10.1007/s00500-018-3546-6.

Kumar, V., & Kaur, A. (2020). Binary spotted hyena optimizer and its application to feature selection. Journal of Ambient Intelligence and Humanized Computing, 11(7), 2625-2645. https://doi.org/10.1007/s12652-019-01324-z.

Li, S., Chen, H., Wang, M., Heidari, A., & Mirjalili, S. (2020). Slime mould algorithm: A new method for stochastic optimization. Future Generation Computer Systems, 111, 300-323. doi:10.1016/j.future.2020.03.055. https://doi.org/10.1016/j.future.2020.03.055.

Mafarja, M., Aljarah, I., Faris, H., Hammouri, A. I., Ala'M, A.-Z., & Mirjalili, S. (2019). Binary grasshopper optimisation algorithm approaches for feature selection problems. Expert Systems with Applications, 117, 267-286. https://doi.org/10.1016/j.eswa.2018.09.015.

Mafarja, M., Heidari, A. A., Faris, H., Mirjalili, S., & Aljarah, I. (2020). Dragonfly algorithm: theory, literature review, and application in feature selection. Nature-inspired optimizers. Springer.

Mafarja, M., & Mirjalili, S. (2018). Whale optimization approaches for wrapper feature selection. Applied Soft Computing, 62, 441-453. https://doi.org/10.1016/j.asoc.2017.11.006.

Mafarja, M., Qasem, A., Heidari, A. A., Aljarah, I., Faris, H., & Mirjalili, S. (2020). Efficient hybrid nature-inspired binary optimizers for feature selection. Cognitive Computation, 12(1), 150-175. https://doi.org/10.1007/s12559-019-09668-6.

Mirjalili, S., & Lewis, A. (2013). S-shaped versus V-shaped transfer functions for binary particle swarm optimization. Swarm and Evolutionary Computation, 9, 1-14. https://doi.org/10.1016/j.swevo.2012.09.002.

Mirjalili, S., Mirjalili, S., & Lewis, A. (2014). Grey Wolf Optimizer. Advances in Engineering Software, 69, 46-61. https://doi.org/10.1016/j.advengsoft.2013.12.007.

Mirjalili, S., Mirjalili, S. M., & Yang, X.-S. (2014). Binary bat algorithm. Neural Computing and Applications, 25(3), 663-681. https://doi.org/10.1007/s00521-013-1525-5.

Neggaz, N., Houssein, E. H., & Hussain, K. (2020). An efficient henry gas solubility optimization for feature selection. Expert Systems with Applications, 152, 113364. https://doi.org/10.23917/jiti.v20i2.15640.

Ouadfel, S., & Abd Elaziz, M. (2020). Enhanced crow search algorithm for feature selection. Expert Systems with Applications, 159, 113572. https://doi.org/10.1016/j.eswa.2020.113572.

Papa, J. P., Rosa, G. H., de Souza, A. N., & Afonso, L. C. (2018). Feature selection through binary brain storm optimization. Computers & Electrical Engineering, 72, 468-481. https://doi.org/10.1016/j.compeleceng.2018.10.013.

Pashaei, E., & Aydin, N. (2017). Binary black hole algorithm for feature selection and classification on biological data. Applied Soft Computing, 56, 94-106. https://doi.org/10.1016/j.asoc.2017.03.002.

Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization. Swarm Intelligence, 1(1), 33-57. https://doi.org/10.1007/s11721-007-0002-0.

Rodrigues, D., Yang, X.-S., De Souza, A. N., & Papa, J. P. (2015). Binary flower pollination algorithm and its application to feature selection Recent advances in swarm intelligence and evolutionary computation (pp. 85-100): Springer. https://doi.org/10.1007/978-3-319-13826-8_5.

Santana Jr, C. J., Macedo, M., Siqueira, H., Gokhale, A., & Bastos-Filho, C. J. (2019). A novel binary artificial bee colony algorithm. Future Generation Computer Systems, 98, 180-196. https://doi.org/10.1016/j.future.2019.03.032.

Sayed, G. I., Hassanien, A. E., & Azar, A. T. (2019). Feature selection via a novel chaotic crow search algorithm. Neural Computing and Applications, 31(1), 171-188. https://doi.org/10.1007/s00521-017-2988-6.

Shayanfar, H., & Gharehchopogh, F. (2018). Farmland fertility: A new metaheuristic algorithm for solving continuous optimization problems. Applied Soft Computing, 71, 728-746. https://doi.org/10.1016/j.asoc.2018.07.033.

Sihwail, R., Omar, K., Ariffin, K. A. Z., & Tubishat, M. (2020). Improved harris hawks optimization using elite opposition-based learning and novel search mechanism for feature selection. IEEE Access, 8, 121127-121145.

Taradeh, M., Mafarja, M., Heidari, A. A., Faris, H., Aljarah, I., Mirjalili, S., & Fujita, H. (2019). An evolutionary gravitational search-based feature selection. Information Sciences, 497, 219-239. https://doi.org/10.1016/j.ins.2019.05.038.

Thaher, T., Heidari, A. A., Mafarja, M., Dong, J. S., & Mirjalili, S. (2020). Binary Harris Hawks optimizer for high-dimensional, low sample size feature selection Evolutionary machine learning techniques. Springer.

Tu, J., Chen, H., Wang, M., & Gandomi, A. (2021). The Colony Predation Algorithm. Journal of Bionic Engineering, 18(3), 674-710. https://doi.org/10.1007/s42235-021-0050-y.

Tubishat, M., Ja'afar, S., Alswaitti, M., Mirjalili, S., Idris, N., Ismail, M. A., & Omar, M. S. (2021). Dynamic salp swarm algorithm for feature selection. Expert Systems with Applications, 164, 113873. https://doi.org/10.1016/j.eswa.2020.113873.

Wang, G. (2018). Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems. Memetic Computing, 10(2), 151-164. https://doi.org/10.1007/s12293-016-0212-3.

Wang, G., Deb, S., & Coelho, L. (2015). Elephant Herding Optimization. 2015 3rd International Symposium on Computational and Business Intelligence (ISCBI). https://doi.org/10.1109/iscbi.2015.8.

Wang, G., Deb, S., & Coelho, L. (2018). Earthworm optimisation algorithm: a bio-inspired metaheuristic algorithm for global optimisation problems. International Journal of Bio-Inspired Computation, 12(1), 1. https://doi.org/10.1504/ijbic.2018.093328.

Wang, G., Deb, S., & Cui, Z. (2019). Monarch butterfly optimization. Neural Computing and Applications, 31(7), 1995-2014. https://doi.org/10.1007/s00521-015-1923-y.

Wang, J., Khishe, M., Kaveh, M., & Mohammadi, H. (2021). Binary Chimp Optimization Algorithm (BChOA): a New Binary Meta-heuristic for Solving Optimization Problems. Cognitive Computation, 13(5), 1297-1316. https://doi.org/10.1007/s12559-021-09933-7.

Yang, X., & Hossein Gandomi, A. (2012). Bat algorithm: a novel approach for global engineering optimization. Engineering Computations, 29(5), 464-483. https://doi.org/10.1108/02644401211235834.

Yang, Y., Chen, H., Heidari, A., & Gandomi, A. (2021). Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts. Expert Systems with Applications, 177, 114864. https://doi.org/10.1016/j.eswa.2021.114864.

Zhang, X., Xu, Y., Yu, C., Heidari, A. A., Li, S., Chen, H., & Li, C. (2020). Gaussian mutational chaotic fruit fly-built optimization and feature selection. Expert Systems with Applications, 141, 112976. https://doi.org/10.1016/j.eswa.2019.112976.

Zhang, Y., & Jin, Z. (2020). Group teaching optimization algorithm: A novel metaheuristic method for solving global optimization problems. Expert Systems with Applications, 148, 113246. https://doi.org/10.1016/j.eswa.2020.113246.

Zhang, Y., Wang, S., Phillips, P., & Ji, G. (2014). Binary PSO with mutation operator for feature selection using decision tree applied to spam detection. Knowledge-Based Systems, 64, 22-31. https://doi.org/10.1016/j.knosys.2014.03.015.

Credit author statement

All authors designed the model and the computational framework and analysed the data and wrote the manuscript.

All Authors

Declaration of interests

☒The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Highlights

- Feature selection method based on Group Teaching Optimization Algorithm is proposed.
- Two novel operators are developed to increase the exploration and exploitation.
- The student phase is enhanced to increase the population diversity and exploitation.
- The teacher allocation phase is enhanced to increase the convergence rate.
- The superiority of the proposed algorithm is indicated over 30 benchmark datasets.