A Novel Approach to Data Clustering based on Self-Adaptive Bacteria Foraging Optimization

Tanmoy Singha[1], Rudra Sankar Dhar[2], Joydeep Dutta[3], Arindam Biswas[4]
Department of Electronic and Communication Engineering, National Institute of Technology, Aizawl, Mizoram, India[1, 2]
Department of Computer Science & Engineering, Siliguri Institute of Technology, Sukna, West Bengal[3]
School of Mines and Metallurgy, KNU, Asansol, West Bengal, India[4]

***Abstract*—Data clustering reduces the number of data objects by grouping similar data objects together. In this process, data are divided into valuable groups (clusters) or expressive without at all previous information. This manuscript represents a different clustering algorithm based on the technique of the adaptive strategy algorithm known as Self-Adaptive Bacterial Foraging Optimization (SABFO). It is a streamlining strategy for bunching issues where a cluster of bacteria forages to converge to definite locations as ultimate group communities by limiting the fitness function. The superiority of this method is assessed on numerous famous benchmark data sets. In this paper, the authors have compared the projected technique with some well-known advanced clustering approaches: the k-means algorithm, the Particle Swarm optimization algorithm, and the Fitness-Based Adaptive Differential Evolution (FBADE) Scheme. An experimental finding demonstrates the usefulness of the projected algorithm as a clustering method that can operate on data sets with different densities, and cluster sizes.**

Keywords—Data clustering; Self-Adaptive Bacterial Foraging Optimization (SABFO); Particle Swarm Optimization (PSO); FBADE scheme; the k-means algorithm and the classical BFO

I. Introduction

A method of analyzing and clustering unlabelled datasets is known as unsupervised machine learning. It is possible to find out the unknown patterns or data groupings using these algorithms without the involvement of human action. In the domain of cross-selling strategies, client segmentation image recognition, etc. The authors can employ the above strategies to find underlying patterns. It also enables us to discover similarities and differences in information. In short, a type of machine learning called unsupervised learning involves training models using unlabelled datasets and allowing them to act upon them without supervision. As opposed to supervised methods, clustering is an unsupervised method that works with datasets that do not have outcomes (target) variables or information about associations among explanations. The clustering algorithm is the key to data analysis and identifying groups (natural clusters). As a result of this process, similar data points are identified and grouped. Clusters make it easier to characterize the attributes of distinct entities. Users can then shift data and analyze certain categories as a result of this. Clustering allows organizations to address distinct client segments based on their attributes and similarities. This aids in profit maximization. If the dataset has too many variables, it can aid in dimensionality reduction. Irrelevant clusters can be detected and deleted from the dataset more easily.

For more than two decades, clustering has taken particular interest among scientists and researchers to apply in versatile domains. Researchers are continuously trying to develop better approaches to clustering.

In order to build knowledge-driven decisions, data mining provides upcoming behaviors that businesses can predict [1]. A clustering technique deals with discovering a structure in an unlabelled collection of data through unsupervised learning [2]. A data cluster is used in order to divide the enormous number of objects into smaller groups, so the objects with similar characteristics are clustered together, and the ones with dissimilar characteristics are in different groups [3]. A cluster is considered too many when it exceeds three, clustering becomes NP-complete, and it is, therefore, challenging to develop a well-organized clustering technique [4]. The clustering problem can be useful for segmenting images [5], clustering documents [6], predicting diseases [7], wireless-related sensors networks [8] analyzing common networks [9], identifying the traffic in the network [10], retrieving information [11], and marketing [12].Partitional clustering is being used in a numbers of real-life applications. It is an algorithm for dividing data objects into small groups based on defined criteria called the distance between them. Data samples are dispersed from one group to another iteratively according to the number of groups determined prior to implementation. To begin, it makes a set of partitions based on a definite measure. Data samples are divided into corresponding groups according to their centres [13]. K-means is the utmost widespread and fashionable algorithm among the partitional clustering algorithms as it is more practical and effective when handling heavy amounts of data. This algorithm's main drawbacks are its sensitivity to the initial cluster centres, inability to find global minima, and convergence to the local optima. Research on the improved BFO algorithm found that the algorithm has some limitations, including a fixed chemotactic step size and feeble bacterial connections.

As a result of the static chemotactic step size, it is problematic to strike the right balance among exploration and exploitation. Secondly, the feeble connection between bacteria shows a poor random city in chemotaxis. As a result of these two drawbacks, the bacteria community will search for a compound multimodal solution set at a local level rather than at a global level when compared to a global convergence.

An approach using self-adaptive BFO (SABFO) is proposed in this paper as one of the novelty of this article. It improves the classical algorithm hypothetically in two ways. By leveraging bacterial search state features, the self-adaptive swimming process overcomes the traditional drawback caused by fixed step sizes. This paper extracts and calculates three vital qualities of the bacterial search state, namely the variety of the population, the number of iterations, and the mean fitness function.

The aim of this manuscript is to propose a new approach to clustering optimization technique, namely SABFO, which will provide performance optimization as the source of data grouping. A new perspective for solving NP-hard clustering problems is provided by Bacterial Foraging Clustering, a global optimization-based technique rather than high speed local search. At the same time, it's a new version of the Bacterial Foraging Optimization technique. In this proposed algorithm there is no need to select the centroid or center required to be chosen in the primary steps.

Further, the proposed algorithm aims to overcome two drawbacks of conventional algorithms:

1) The projected clustering technique achieved a high degree of accuracy as compared with other algorithms.

2) High-dimensional data can be processed resourcefully with the proposed algorithm.

The rest of this manuscript is planned in a subsequent way. In Section II, the literature review is introduced. Section III illustrates the preliminary knowledge of the BFO algorithm and optimization-based clustering. Section IV illustrates fully the entire method of the recommended SABFO-Clustering algorithm. In Section V, the authors present the numerical illustration for the datasets used in this paper. Sections VI and VII summarizes the results and discussion, respectively. Finally, Section VIII shows the manuscript's conclusion and future work.

II. Literature Review

The foraging performance of 18 Escherichia coli in the human being intestinal tract was studied, Passino [14] proposed a bacterium foraging optimization (BFO) algorithm in 2002 for optimizing 17 problems. Despite the 19 BFO algorithm's superiority over several other algorithms, 20 its convergence speed 21 and global search capability need to be improved, because it is extremely simple to fall into local optimal outcomes and convergence is slow.

Different clustering algorithms in recent years have been projected such as segmentation, density based hierarchical, grid-based, and model-based. By using partitioning, the authors can create partitions based on a number of criteria. The pattern belongs to only one cluster when using hard Partitional clustering. Clustering by fuzzy rules extends this notion by allowing patterns to belong to more than one cluster.

During the last few years, the BFO algorithm has often been combined 49 with other algorithms in various fields, Ofosu et al. In 50 [13], the proportional integral and derivative controllers 54 were unable to overcome the difficulties encountered in obtaining optimal PI gains 51 for fuzzy-PI controllers.

An optimal allocation model based on risk has been proposed by Xiong and et al. [15] and a multi-objective optimization57 problem has been solved by merging gradient particle, 58 swarm optimization with bacterial optimization reduces the risks associated with distributed 60 generation and facilitates the advancement of and implementation of distributed generation.

TABLE I. Literature Review

Author	Year	Technique introduced	Results
Tripathy, M and et.al [17]	2006	Enhanced Bacteria Foraging Optimization	Retained least cost
Li, M.S and et.al [18]	2007	Bacteria Foraging Algorithm varying population	Quorum sense, proliferation
Biswas, A and et.al [19]	2007	Genetic algorithm	Global optimization
Korani, W and et.al [20]	2008	PSO	Proportional – Integral – Derivative controller tuning
Dasgupta, S. and et.al [21]	2009	Micro Bacteria Foraging Optimization	Smaller population
Chen, H and et.al [22]	2009	Cooperative Bacteria Foraging Optimization	Explicit decomposition of search space
Dasgupta, S and et.al [23]	2009	Adaptive Bacteria Foraging Optimization	Varying chemotactic steps
Chen, H and et.al [24]	2010	Multi colony BFO	Several colonies
Kim, D.H and et.al [25]	2011	Genetic algorithm	Proportional – Integral – Derivative controller tuning
Gollapudi, S.V.R.S and et.al [26]	2011	PSO	Resonant frequency of rectangular micro strip antenna
Okaeme, N.A and et.al [27]	2013	Genetic algorithm	Automated investigational control design
Abd-Elazim, S.M and et.al [28]	2013	PSO	Power system stabilizers illustration
Mandeep Kaur and et.al [29]	2018	MOBFOA	Comparative study with other algorithm
Lv, X and et.al [30]	2018	IBFO	Machine Learning Framework
Huang Chen and et.al [31]	2020	SCBFO	Demonstration of CEC 2015 benchmark test set
Yufang Dan and et.al [32]	2021	BFO	Dynamic, multi-objective optimization, and complicated constrained optimization
Bo Yang and et.al [33]	2022	Discrete BFO	Unveiling global communities in networks
Sandeep Gogula and et.al [34]	2023	BFO	Size of the DGs, losses in active and reactive power flow

Guo and Zhou [16] employed the trapezoid quadrature formula 65 in combination with the 64 BFO techniques to compute integrals since 64 integrable functions have many primitive functions that aren't elementary. Table I represents the tabular form of literature review with year, Technique and results used by the authors.

III. Preliminaries

A. BFO based Clustering

The process of clustering is a data mining method that involves classifying objects without any prior knowledge (clusters).It is possible to formalize the clustering problem as follows, given a sample data set X=(x_(1,) x_(2 ,)……x_(n)),It is possible to formalize the clustering problem as follows, given a sample data set ,determine a partition of the objects into K clusters which satisfies:

$$\cup_{i=1}^{k} C_i = X; \qquad (1)$$

$$C_i \cap C_j \left\{ = \emptyset \ , \ i,j = 1,2, \dots \dots .. k; \ \ i \neq j \right. \qquad (2)$$

$$C_i \ \neq \theta \quad i = 1,2, \dots \dots .. k$$

In the mathematical point of view, cluster C_i can be obtained by:

$$\begin{cases} |C_I = \{x_j | \|x_j - z_i\| \leq \|x_j - z_p\|, x_j \in X\} \\ q \neq 1, q = 1,2 \dots \dots \dots .. k \\ z_i = \frac{1}{|C_i|} \sum_{C_j \in C_i} x_j, i = 1,2, \dots \dots .. K \end{cases} \qquad (3)$$

Where,‖.‖ Signifies the length between of any two data points in the trial set, and z_i =the centre of cluster C_i .

B. The Classical BFO Algorithm

BFO algorithm is enlivened with a movement known as "chemotaxis" showed by bacterial foraging ways of behaving. Motile bacteria like E. coli and salmonella impel themselves by the turn of the flagella. An organic entity swimming or running forward is caused by the flagella turning counterclockwise, while a bacterium that makes a clockwise pivot tumble about with haphazard motion and swims once again. The bacterium can look for nutrients in any direction by switching among "swim" and "tumble" motions. The bacterium begins to swim more often as it gets closer to a nutritional gradient. Bacteria move away from some nourishment to search for further, resulting in tumbling, hence direction changes. Chemotaxis, in its simplest form, involves bacteria swimming and tumbling to reach advanced concentrations of food.

C. Bacterial Foraging Optimization

The three main mechanisms that make up the classical BFO system are chemotaxis, reproduction, and elimination-dispersal. Here are quick summaries of each of these processes:

1) The basic chemotaxis: Chemotaxis for bacteria is the course of the accumulation to nutrient-enriched regions. Bacterial movement designs incorporate both tumbling and swimming. Flips are the unit step lengths that bacteria take when moving in any direction. The extent to which adjustments are necessary determines whether the new position is more attractive than the opposite position. Then, the bacterium will keep up shifting in a more excited propensity for a couple of steps till the limit for variation is not any more shut. The enhanced method will be

$$Q_i\,(j+1,k,l) = Q_i\,(j,k,l) + \frac{\Delta_i}{\sqrt{\Delta^T(i)\Delta_i}}\,C(i)n \qquad (4)$$

Here, $Q_i\,(j+1,k,l)$symbolize the ith bacterium at the jth chemotactic, kth denotes the reproductive, and lth represented the elimination dispersal steps. $C(i)n$ is denoted as the trend step length of bacteria i in a random direction and Δ lies between −1 and 1 as a random vector.

2) Swarming: The chemotactic behavior of bacteria is not limited to searching for food individually but also includes both gravitation and repulsion between them in the foraging process. A bacteria's attractive information causes it to move to the center of the population, thus fetching the bacteria closer together. Yet, the bacteria's repulsion information keeps them at a distance from each other at the same time.

3) Reproduction: Eventually, bacteria with weak feeding abilities will be removed, while bacteria with robust feeding capabilities evolve to breed offspring to continue the population size. This process follows the usual method of survival of the fittest. The authors proposed a reproduction operation based on simulating this phenomenon. A chemotaxis operator performed by S/2 bacteria eliminated bacteria with poor fitness and let those with higher fitness self-replicate in S-sized populations.

A completed reproduction operation ensures that the offspring inherits the superior characteristics of the parents, and it also results in the protection of the good individuals and the acceleration of the progress towards an optimal global outcome. Fig. 1 represents the basic structure of the Bacteria Foraging Optimization algorithm.

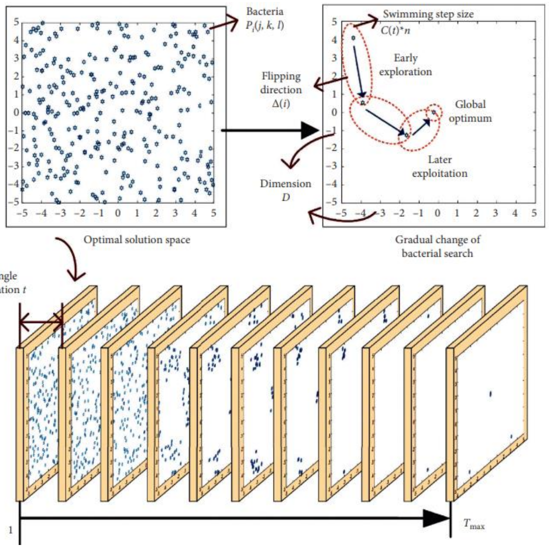

Fig. 1. The basic organization of the Bacteria Foraging Optimization algorithm.

4) Elimination: It is important not to rule out the possibility that unexpected conditions might cause bacteria to die or migrate to a new location during bacterial foraging. It has been proposed to model this phenomenon by simulating elimination-dispersal operations.

IV. Proposed Self-Adaptive BFO (SABFO)

Chemotaxis is a crucial tool in exploring and exploitation of the BFO algorithm during the search for the optimization space. The consequence of chemotaxis depends on the size of the steps and the direction the swimmer flips.

Two improvements are proposed in this paper to get better performance of the BFO algorithm, including extracting and calculating the features of the search state and increasing bacteria's communication. The SABFO algorithm provides a novel BFO algorithm to design dynamic self-adaptive swimming and flipping motions for bacterial cells with these two improvements.

It seems that the method of calculating swimming step size depends on the single swimming step size, C(t), as well as the quantity of chemotaxis, n, as indicated by the above researcher. An algorithm's performance can be adjusted to the ebb and flow state of search based on the size of the swimming steps. At the point when the pursuit state is in the beginning phase, the calculation needs the investigation capacity for worldwide pursuit; then, at that point, in the later stage, the abuse capacity is expected for nearby turn of events.

For various optimization issues, the difference in the BFO search state is likewise unique. In the meantime, on the grounds that the chemotaxis method is nonlinear and it is very complex to such an extent that the progress from the worldwide investigation to the nearby double-dealing can't be essentially depicted and separated by the way of logical conditions.

The proposed paper separates the three components of the BFO in every search state so that the authors can better understand the unique change of the BFO algorithm to the suitable chemotaxis swimming, including iteration, population diversity, and mean fitness. Fig. 2 represents the bacterial population position of BFO Algorithm.

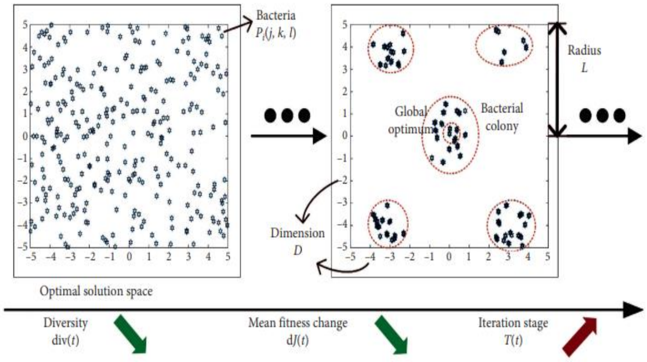

Fig. 2. The bacterial population position of BFO algorithm.

The population diversity of bacteria refers to where it is dispersed. A wider range of bacteria dispersing will increase population diversity, and vice versa. As part of the chemotaxis process of BFO, this manuscript measures the bacterial colony's population diversity.

$$div(t) = \frac{1}{D*S}\, g\sqrt{\sum_{i=1}^{S}\left(\frac{Q_i(j,k,l) - \overline{Q_i(j,k,l)}}{|L|}\right)} \tag{5}$$

Where div (t) is the range between [0, 1], L denotes the solution space's elongated radius. This method measures the distance between the center of each bacterium and the solution space, regardless of the amount or dimension of the bacteria.

The iteration of the BFO algorithm is communicated through a boundary T, which is characterized as an articulation in the reach of [0,1] in equation 6, where t and T_{max} address the record of the current chemotaxis and the most extreme emphasis, separately. Hence, the meaning of parameter T is for the most part suitable to various algorithms regardless of how the parameters, the aspect, with the arrangement space which are programmed into the algorithms:

$$T(t) = \frac{t}{T_{max}} \tag{6}$$

The modification in the mean fitness function is in two chemotaxis processes, where dJ is primarily calculated as one of the essential values for examining the BFO algorithm. To provide a common explanation, the difference in the mean fitness dJ is characterized in the per-unit structure inside [−1,1] as follows:

$$dJ(t) = \frac{J(t) - J(t-1)}{J_{Max} - J_{Min}} \tag{7}$$

Where, J_{Max} andJ_{Min} denotes the maxima and minima of the fitness function, respectively. Fig. 3 shows the flowchart of the proposed algorithm SABFO.

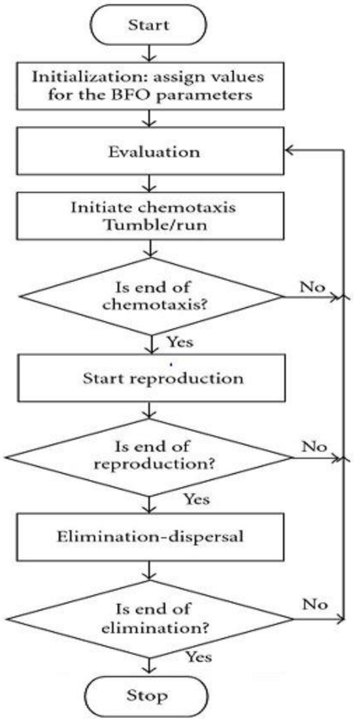

Fig. 3. The flowchart of the proposed algorithm.

So, in this proposed manuscript, the 03 most important variables are the population diversity, the number of iterations and the bacteria's mean fitness value is used as input, which is intended to examine the pursuit status of the algorithm in these manuscripts. As the chemotaxis will be changed as per accompanying with the swim processes are as follows:

$$\begin{cases} n(t+1) = n(t) + dn(t) \\ C(t+1) = C(t)gC_M(t) \end{cases} \quad (8)$$

Where, the 02 output variables are the bacterial swimming movement increases $dn(t)$ [0.01, 1].as well as the bacterial swimming step multiple is C(t) (0, 1).

Another important operation of the BFO algorithm is flipping the bacteria. The direction of chemotaxis is determined by the extremum of each bacterium when swimming. Despite its benefits to the randomness of the search, this method results in a slower search because of blocked information among the bacteria. Therefore, BFO algorithms with suffer from the drawback of tumbling into the local optimum.

In order to resolve the issue, mentioning individuals' data exchange information strategy is used in BFO. So, the BFO algorithm is updated and the flipping variable is given by

$$\Delta_{t+1}(i) = wg\Delta_t(i) + C_1R_1\left(P_{local} - P_{N_c}\right) + C_2R_2\left(P_{global} - P_{N_c}\right) \quad (9)$$

By adjusting the co-efficient, the chemotaxis process is used in the above-mentioned equation.

Where w, denotes the chemotaxis inertia of the bacteria at a specific distance.

During bacterial chemotaxis, C1 signifies the amount at which each bacterium travels toward its distinct optimal value P_{local}, whereas C2 records the global optimum value P_{global} for all bacteria. For improving the randomness of bacterial flipping and enhancing search ability, R1 and R2 are random the values between 0 and 1. The flowchart of the algorithm is shown in the Fig. 3.

V. Numerical Illustration

In order to compare this proposed approach to Self-Adaptive BFO, five real-life data are Iris [35], Glass [35], breast cancer [35], Wine [35] and Vowel Dataset [35] were used in this proposed paper.

Real- Life Data Sets

- Iris Data: It comprises of three distinct types of iris blossom.
- Glass: The information was examined from six dissimilar kind of glass.
- Breast cancer: It comprises of 9 applicable highlights.
- Wine Data set: It is the outcome of a chemical analysis of wine. This analysis is resolved with the quantities of 13 constituents shown in every one of the three kinds of wine.
- Vowel Dataset: It comprises of 871 Indian Telugu vowel sounds.

The authors have used the following real-life datasets in this proposed paper. Table II represents the data set used in the manuscript.

TABLE II. The Datasets Used

Real-Life Dataset	n	D	K
Iris plants [35]	150	4	3
Glass[35]	214	9	6
Wisconsin breast Cancer data set[35]	683	9	2
Wine[35]	178	13	3
Vowel Dataset[35]	871	3	6

Where, n represents number of data points. D represents number of features and K represents no. of Cluster.

Similarly Table III represents the value of parameter used in the article.

TABLE III. Value of Parameter

Algorithm Name	Parameter Name	Value
FBADE	Population size	10*dim
	Crossover	0.9
	Mutation	0.8
	K_{max}	20
	K_{min}	2
K Mean	Population size	50
	μ_c	8
	μ_m	0.001
	K_{max}	20
	K_{min}	2
PSO	Population size	100
	Inertia Weight	0.72
	C_1, C_2	1.494
	$P_{initial}$	0.75
	$K_{maximum}$	20
	$K_{minimum}$	2
SABFO	Population size (S)	50
	N_C (No. of Chemotactic Steps)	100
	N_S (Length of One swim)	4
	N_{re}(No. of reproduction steps)	4
	N_{ed} (No. of elimination dispersal events)	2
	P_{ed} (Probability of elimination dispersal events)	0.25

VI. Result Analysis

Three performance metrics have been used to compare the SABFO algorithm with other evolutionary algorithms as state-of-the-art clustering techniques:

1) Performance metrics in the CS and DB domains as well as the number of misclassified items for each dataset;

2) Finding the optimal number of clusters; and

3) Computing time.

It is first necessary to measure the fair time of stochastic algorithms like PSO, FBADE, SABFO, and K Mean in order to compare their speed. Meanwhile the algorithms perform a dissimilar amount of work within their inner loops, as well as having different populations, the number of runs or generations cannot be used as a time measurement. Thus the authors choose to calculate computation time based on the number of fitness function evaluations (FEs) rather than the number of generations and iterations. When function complexity increases, counting the FEs is a reliable gauge of runtime complexity because it corresponds strongly with actual processor time.

Generally, two successive runs of four competing algorithms do not match because they are stochastic. As a consequence, the authors conducted 50 independent runs of each algorithm using different seeds. Based on the 40 runs, each result is expressed as a mean and standard deviation. As the various leveled agglomerative calculation utilized here, utilizes no developmental strategy, the amount of function evaluations isn't applicable to this technique. In this algorithm, the authors use the Ward updating equation to efficiently calculate cluster distances given the number of clusters for each problem.

Depending on the clustering validity measure used, any of the four evolutionary clustering algorithms will perform well. A CS measure-based fitness function is used in one set of experiments, while a DB measure-based fitness function is used in the other set of experiments. Four partitional clustering algorithms have been evaluated in terms of CS and DB calculation against the average-link metric based hierarchical method for each dataset.

This algorithm was run in Matlab 2010 under Windows 11 using an Intel Core i5 computer having 3.60 GHz speed and 8 GB of RAM.

The SABFO algorithm continues to offer superior clustering accuracy to each of the other three competitors as shown in Table IV. Tables IV and V represent the first four evolutionary algorithms (using the CS measure), mean classification error and standard deviation over nominal partitions were determined over 40 independent runs.

TABLE IV. 10^6 FUNCTION EVALUATIONS (FES) WITH CLUSTER STRICTNESS (CS)

Name of the Dataset	Algorithm	Avg No. of clusters found	Value of CS calculated	Mean Intra cluster Distance	Mean Inter cluster Distance
BreastCancer	SABFO	**2.26±0.00**	**0.4623±0.033**	**4.2356±0.143**	**3.2489±0.138**
	PSO	2.13±0.0587	0.4878±0.009	4.7845±0.356	2.3521±0.021
	K Mean	2.00±0.0079	0.5098±0.015	4.8879±0.904	2.3857±1.699
	FBADE	2.06±0.0232	0.4854±0.359	4.5944±0.599	2.8977±1.345
	Classical BFO	2.15±0.0261	0.8984±0.381	4.5644±0.546	3.0625±1.455
Vowel	SABFO	**5.72±0.0641**	**0.9068±0.046**	**1399.96±0.692**	**2698.58±0.112**
	PSO	7.25±0.0183	1.1827±0.431	1482.51±3.973	1923.93±1.154
	K Mean	5.05±0.0075	1.8978±0.897	1485.13±12.235	1921.38±0.742
	FBADE	7.50±0.0569	1.0844±0.067	1493.72±10.833	2434.45±1.213
	Classical BFO	6.66±0.0895	1.2335±0.048	1499.96±0.956	2698.58±0.112
Glass	SABFO	**6.04±0.0139**	**0.3221±0.456**	**563.247±134.2**	**853.62±9.044**
	PSO	5.89±0.0093	0.7532±0.073	599.535±10.34	889.32±4.233
	K Mean	5.82±0.0346	1.4743±0.236	594.673±30.62	869.93±1.789
	FBADE	5.59±0.0754	0.6999±0.643	608.787±20.92	891.82±4.945
	Classical BFO	5.89±0.0654	0.7506±0.725	598.852±166.3	890.89±8.250
Iris	SABFO	**3.198±0.0382**	**0.6548±0.097**	**3.106±0.033**	**2.3941±0.027**
	PSO	2.23±0.0443	0.7361±0.671	3.6516±1.195	2.2104±0.773
	K Mean	2.35±0.0985	0.7282±2.003	3.5673±2.792	2.5058±1.409
	FBADE	2.50±0.0473	0.7633±0.039	3.9439±1.874	2.1158±1.089
	Classical BFO	2.65±0.0752	0.7531±2.003	3.6689±1.562	2.2515±1.233
Wine	SABFO	**3.19±0.0391**	**0.8989±0.032**	**4.041±0.002**	**3.1399±0.078**
	PSO	3.03±0.0253	1.7899±0.037	4.787±0.184	2.6113±1.637
	K Mean	2.95±0.0112	1.5842±0.328	4.163±1.929	2.8058±1.365
	FBADE	3.50±0.0143	1.7964±0.802	4.949±1.232	2.6118±1.384
	Classical BFO	3.65±0.0562	1.6998±0.056	4.655±0.095	2.922±1.563

TABLE V. MEAN CLASSIFICATION ERROR

Dataset	Mean Classification Error				
	SABFO	**PSO**	**K Mean**	**FBADE**	**Classical BFO**
Breast Cancer	21.98±0.28	27.01±1.25	29.00±1.55	29.15±0.50	26.00±0.00
Vowel	413.88±3.08	451.58±5.98	471.69±6.89	474.72±4.25	496.00±0.00
Glass	91.51±0.19	102.1±0.68	98.21±0.08	105.36±0.54	111.00±0.00
Iris	2.35±0.00	4.15±0.0	5.00±0.00	3.96±0.00	4.00±0.00
Wine	36.52±0.0	98.4±1.09	100.24±1.05	114.50±1.53	134.00±0.00

TABLE VI. DB VALUES AT THE PREDEFINED CUT-OFF VALUE WERE CALCULATED AFTER 50 INDEPENDENT RUNS, AND THE MEAN CLASSIFICATION ERROR

Name of the Dataset	Name of the Algorithm	Mean no. of FE's required	DB Cutoff Value	Mean Intra cluster Distance	Mean Inter cluster Distance
Iris	SABFO	**504783.45**±12.65	0.8	**3.9928±0.029**	**2.1029±0.842**
	PSO	679084.75±16.57		3.7852±1.842	1.7641±0.439
	K Mean	790865.90±10.21		4.4587±3.782	1.9383±1.307
	FBADE	658796.3		4.0393±1.5	1.6278±1.6
Wine	SABFO	**464653.35**±5.50	6	**4.8292±0.732**	**3.0219±0.069**
	PSO	486885.85±2.85		5.1472±0.472	2.1161±1.623
	K Mean	598743.35±8.09		4.9383±1.722	2.9121±0.353
	FBADE	477869.95±8.12		4.7531±2.043	2.8158±0.389
Breast-Cancer	SABFO	424732.30±8.93	0.9	5.4489±0.342	3.0234±0.683
	PSO	467854.60±10.12		5.2885±0.552	2.0124±1.596
	K Mean	678874.90±7.82		6.8832±0.733	2.1637±1.458
	FBADE	**418765.55±1.23**		**5.8684±0.467**	**1.9235±0.164**
Vowel	SABFO	**435743.05±2.65**	3	**1544.92±0.834**	**2081.31±0.679**
	PSO	556865.00±4.26		1652.58±2.341	1264.87±3.069
	K Mean	575854.65±1.29		1582.55±7.332	1989.38±7.734
	FBADE	546859.60±2.05		1608.22±5.866	1604.43±1.674
Glass	SABFO	**506754.00**±12.27	2	**132.757±15.8**	**13.46±2.54**
	PSO	569787.95±10.83		154.564±39.6	13.56±2.65
	K Mean	687678.75±10.97		155.856±24.7	10.42±4.69
	FBADE	527585.35±7.50		178.809±30.3	10.21±1.09

CS and DB index values were reduced by the SABFO within a minimum number of function evaluations in the majority of cases, as shown in Tables VI. According to Table VI, SABFO continues to provide superior clustering accuracy to the other three competitors. Entries of Statistically significant differences between SABFO and its competitors are evident in Table VI, only for breast cancer, FBADE yield a lower DB value than SABFO. Table VII shows the mean classification error and standard deviation of the different data set.

Table VIII represent the first four evolutionary algorithms (using the DB measure), mean classification error and standard deviation over nominal partitions were determined over 40 independent runs.

TABLE VII. MEAN CLASSIFICATION ERROR AND STANDARD DEVIATION

Dataset	**Mean Classification Error**				
	SABFO	**PSO**	**K Mean**	**FBADE**	**Classical BFO**
Iris	**2.22±0.00**	2.79±0.55	2.75±0.08	2.74±0.00	3.14±0.00
Wine	**40.15±0.0**	112.5±2.50	118.45±1.77	76.45±0.236	102.22±1.05
Breast Cancer	**26.72±0.25**	30.33±0.48	26.55±0.79	29.00±1.12	29.03±1.09
Vowel	**416.37±7.50**	437.00±3.72	476.58±3.59	478.62±2.69	
Glass	**8.86±0.42**	14.35±0.26	17.98±0.67	15.69±0.85	

TABLE VIII. DB MEASURE-BASED FITNESS FUNCTIONS

Name of the Dataset	**Name of the Algorithm**	**Average Number of clusters found**	**Value of DB calculated**	**Mean Intra cluster Distance**	**Mean Inter cluster Distance**
Iris	SABFO	**3.48±0.0217**	**0.4644±0.029**	**3.1636±0.078**	**2.8389±0.678**
	PSO	2.28±0.0598	0.6677±0.008	3.8536±0.122	2.2548±0.034
	K-Mean	2.32±0	0.7269±0.0	3.8428±0.076	2.1438±0.020
	FBADE	2.51±0.0089	0.5825±0.069	3.8879±0.089	2.0358±0.058
	Classical BFO	2.96±0.008	0.8674±0.00	3.8098±0.00	2.2857±0.00
Wine	SABFO	**3.25±0.0931**	**3.0432±0.021**	**4.4212±0.096**	**3.1029±0.047**
	PSO	3.05±0.0024	4.3432±0.232	4.8668±0.154	2.6113±1.635
	K-Mean	2.95±0.0173	5.3424±0.343	5.1312±1.342	2.7565±2.128
	FBADE	3.50±0.0143	3.3923±0.092	4.263±1.907	2.8158±1.786
	Classical BFO	2.99	5.7206±0.00	4.982±0.00	2.5009±0.00
Breast Cancer	SABFO	2.48±0.0653	0.5102±0.007	4.5564±0.024	3.1020±0.068
	PSO	2.50±0.0621	0.5754±0.073	4.9232±0.373	2.2684±0.063
	K-Mean	2.50±0.0352	0.6328±0.002	6.5541±0.433	1.8032±0.016
	FBADE	**2.10±0.0081**	**0.5199±0.007**	**5.2234±0.042**	**2.0236±0.058**
	Classical BFO	2	0.7634±0.00	5.0098±0.00	2.2817±0.00
Vowel	SABFO	**5.75±0.0241**	**0.9224±0.334**	**1449.12±0.834**	**2289.85±0.163**
	PSO	7.25±0.0562	1.2821±0.009	1500.57±3.748	1747.76±1.764
	K-Mean	5.05±0.0561	2.9482±0.028	1573.23±4.675	2271.89±1.222
	FBADE	7.50±0.0819	1.4488±0.075	1498.78±2.725	1962.31±0.993
	Classical BFO	6	3.0581±0.00	1493.98±0.00	2357.62±0.00
Glass	SABFO	**6.05±0.0248**	**1.0092±0.083**	**501.757±4.3**	**893.46±3.32**
	PSO	5.95±0.0193	1.5152±0.073	514.554±9.5	856.00±8.07
	K-Mean	5.85±0.0346	1.8371±0.034	518.903±2.9	852.32±5.43
	FBADE	5.60±0.0446	1.6673±0.004	514.849±3.4	862.21±2.53
	Classical BFO	6	1.8519±0.00	610.033±0.00	895.47±0.00

TABLE IX. MEAN CLASSIFICATION ERROR

Dataset	**Mean Classification Error**				
	SABFO	PSO	K-Mean	FBADE	Classical BFO
Iris	**2.21±0.02**	2.80±0.56	2.78±0.10	3.15±0.07	2.75±0.01
Wine	**41.25±0.01**	112.5±2.50	118.45±1.77	103.20±1.05	58.15±0.08
Breast Cancer	**27.69±0.28**	30.23±0.46	26.50±0.80	29.00±1.09	29.08±0.25
Vowel	**417.39±6.99**	435.00±3.75	473.46±3.57	472.65±2.76	486.65±3.26
Glass	**8.82±0.42**	14.56±0.28	17.98±0.67	15.70±0.89	17.52±0.68

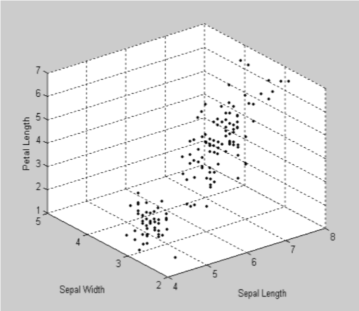

Fig. 4. The 3D plot of the unlabeled Iris data set.

The authors have applied various well-known advanced clustering approaches like the k-means algorithm, the Particle Swarm optimization algorithm, and the Fitness-Based Adaptive Differential Evolution (FBADE) Scheme on the 3D plot of Iris data (Fig. 4). The clustering results are as follows:

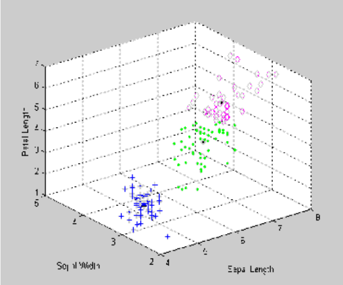

Fig. 5. Clustering of iris data by SABFO.

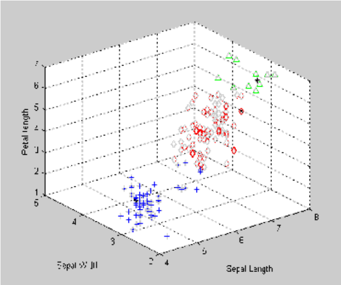

Fig. 6. Clustering of iris data by PSO.

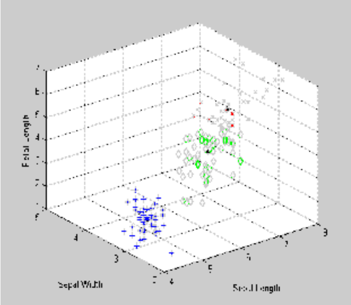

Fig. 7. Clustering of iris data by K-Mean.

Fig. 5 can classify the data set which contain overlapped clusters very efficiently and it also the ability to cluster data sets with high dimension as compared to Fig. 6, 7 and 8.

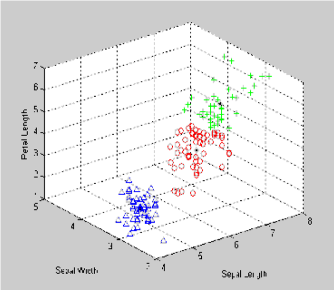

Fig. 8. Clustering of iris data by FBADE.

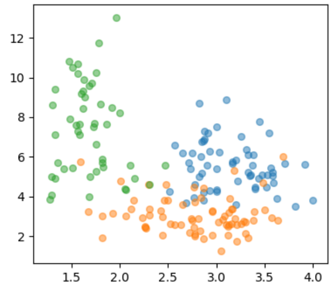

Fig. 9. The 1D plot of the unlabeled Wine data set.

Fig. 9 represents the 1-Dimensional plot of the unlabeled wine data set. The authors have also applied various well-known superior clustering approaches like the k-means algorithm, the Particle Swarm optimization algorithm, and the Fitness-Based Adaptive Differential Evolution (FBADE) Scheme on the 1D plot of Wine Data set. The clustering results are as follows:

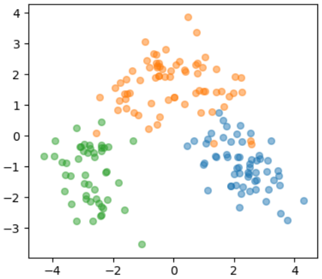

Fig. 10. Clustering of unlabeled Wine data set by SABFO.

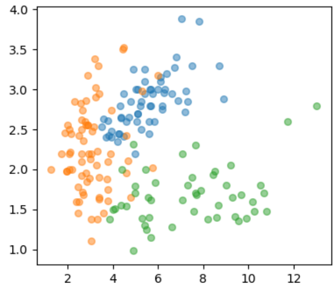

Fig. 11. Clustering of unlabeled Wine data set by PSO.

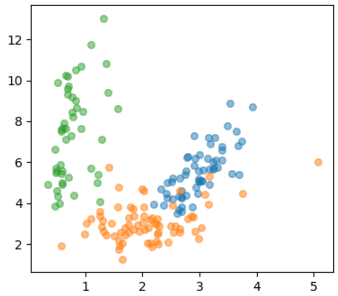

Fig. 12. Clustering of unlabeled Wine data set by K-Mean.

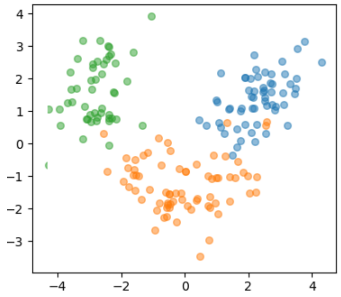

Fig. 13. Clustering of unlabeled Wine data set by FBADE.

Fig. 10 can classify the data set which contain overlapped clusters very efficiently and it also the ability to cluster data sets with high dimension as compared to Fig. 11, 12 and 13.

VII. Discussion

As can be seen in Table IV, the SABFO algorithm continues to offer clustering accuracy that is superior to that of the other three competitors. The first four evolutionary algorithms are shown in Tables IV and V. The mean classification error and standard deviation over nominal partitions were calculated after 40 independent runs using the CS measure.

CS and DB record values were decreased by the SABFO inside a base number of capability assessments in most of cases, as displayed in Tables VI. According to Table VI, SABFO keeps on giving better bunching exactness than the other three contenders. Passages of Genuinely tremendous contrasts among SABFO and its rivals are obvious in Table VI, just for bosom disease, FBADE yield a lower DB esteems than SABFO.

Clustering problems that have several data items, clusters, and overlapping cluster shapes have noticeable performance changes. The clustering accuracy of SABFO is consistently superior to that of its competitors in both Tables IV and V. The FBADE and SABFO methods have two clusters nearly same every time when it's run for the breast cancer dataset, despite having very similar final CS indices. Entries of Statistically significant differences between SABFO and its competitors are evident in Table VI, only for breast cancer, FBADE yield a lower DB value than SABFO.

The results of Tables V and IX indicate that the SABFO produces the fewest misclassified items after clustering. Although all five algorithms demonstrated convincing performance, there were misclassifications in each experiment based on the nominal classification, as expected. In this proposed evolutionary clustering algorithms, the authors found that the fitness values obtained were much better than those obtained from the insignificant classification, which represents that optimization cannot explain through misclassification. As a result, misclassification is caused by underlying expectations in the clustering fitness values (such as clusters' spherical shape), outliers in the dataset, and errors in data collection and nominal solutions. This is indeed not a negative result. Clustering solutions based on statistical criteria and minor classifications can be compared to reveal interesting data points and anomalies. Using a clustering algorithm to pre-analyze data in this way can be very useful.

According to Tables IV and VIII, both the CS and DB indices reached their cut-off values within a minimum number of FE's.

VIII. Conclusion

A SABFO algorithm was proposed in this manuscript to address the fixed step size of the classical BFO algorithm as well as weak correlation among bacteria. Self-adaptive chemotaxis is an adaptation of the self-adaptive swimming technique depends on bacteria's state of search features, combined with enhancement of chemotaxis flipping based on exchange of information.

A comparison of the SABFO algorithm on 05 data sets was conducted by the PSO algorithm, the FBADE algorithm, the K-Mean algorithm and the classical BFO. It was found that SABFO's algorithm is accurate and effective at determining optimal solutions based on the validation results. The SABFO algorithm was also demonstrated to have better exploitation abilities in the future stages and having a more steady search performance.

In brief, the SABFO algorithm has a good stability between exploration and exploitation, which reduces the risks of local convergence. Further it can overwhelm the aforesaid two shortcomings of traditional BFOs.

Additionally, the SABFO algorithm is very stable and performs well when searching. Due to this, SABFO provides an efficient and novel way to accord with complex optimization issues.

In the above table, it appears that all four competitor algorithms terminated with similar accuracy for all the datasets. Based on the proposed algorithm, the CS and DB are found very lowest as per Table IV and VIII. In addition, SABFO successfully found the near-exact number of classes over consecutive iterations (three for iris and Wine data sets).For future researchers, there is a lot of scope to improve the proposed variants that may give much more excellent results on real-world optimization problems.

References

[1] P. P. Mohanty, S. K. Nayak, U. M. Mohapatra, and D. Mishra 2019 A survey on partitional clustering using single-objective metaheuristic approach. Int. J. Innovative Comput. Appl. vol.10 pp.207-226.

[2] A. Abraham, S. Das, and S. Roy 2008 Swarm intelligence algorithms for data clustering in Soft computing for knowledge discovery and data mining Springer pp.279-313.

[3] T. Niknam and B. Amiri, 2010 An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis Appl. Soft Comput. Vol.10 pp.183-197.

[4] G. Sahoo 2017 A two-step artificial bee colony algorithm for clustering Neural Comput. and Appl. vol.28 pp.537-551.

[5] A. Nithya, A. Appathurai, N. Venkatadri, D. Ramji, and C. A. Palagan 2020 Kidney disease detection and segmentation using artificial neural network and multi-kernel k-means clustering for ultrasound images Measurement vol.149.

[6] A. Christy and G. M. Gandhi, 2020 Feature Selection and Clustering of Documents Using Random Feature Set Generation Technique in Advances in Data Science and Management: Springer pp.67-79.

[7] S. Ramasamy and K. Nirmala, 2020 Disease prediction in data mining using association rule mining and keyword based clustering algorithm Int. J. Comput. Appl. vol.42 pp.1-8.

[8] D. K. Kotary and S. J. Nanda, 2020 Distributed robust data clustering in wireless sensor networks using diffusion moth flame optimization Engineering Applications of Artificial Intelligence vol.87 First International Conference on Advances in Physical Sciences and Materials Journal of Physics: Conference Series 1706 (2020) 012163 IOP Publishing doi:10.1088/1742-6596/1706/1/01216310.

[9] S. A. Curiskis, B. Drake, T. R. Osborn, and P. J. Kennedy, 2020 An evaluation of document clustering and topic modeling in two online social networks: Twitter and Reddit Inf. Process. Lett. & Manage. vol.57.

[10] J. Yang, Y. Han, Y. Wang, B. Jiang, Z. Lv, and H. Song, 2020 Optimization of real-time traffic network assignment based on IoT data using DBN and clustering model in smart city Future Gener. Comput. Syst. vol.108 pp.976-986.

[11] P. Bedi and S. Chawla, 2010 Agent based information retrieval system using information scent Int. J. Artif. Intell. vol.3 pp.20-238.

[12] B. Yue, 2020 Topological Data Analysis of Two Cases: Text Classification and Business Customer Relationship Management in J. Phys. Conf. Ser. vol.1550.

[13] A. José-García and W. Gómez-Flores, 2016 Automatic clustering using nature-inspired metaheuristic : A survey Appl. Soft Compt. vol.41 pp.192-213.

[14] K. M. Passino, Biomimicry of bacterial foraging for distributed optimization and control, IEEE Control Systems Magazine (Volume: 22, Issue: 3, June 2002).

[15] R. A. Ofosu, S.I. Kamau, J.N. Nderu, et al., Determination of Optimal PI Gains For Fuzzy-PI Controller Using Bacterial Foraging Algorithm (BFA), IOSR Journal of Electrical and 588 Electronics Engineering 11(2) (2016), 26–33.

[16] D. Guo and J. Zhou, Numerical integration based on bacterial foraging algorithm, Science and Technology Vision 10 555 (2019), 118–120. 556.

[17] Tripathy, M., Mishra, S., Lai, L.L., Zhang, Q.P.: Transmission loss reduction based on FACTS and bacteria foraging algorithm. In: Proceedings of PPSN, pp. 222–231 (2006).

[18] Li, M.S., Tang, W.J., Tang, W.H., Wu, Q.H., Saunders, J.R.: Bacteria foraging algorithm with varying population for optimal power flow. In: Proceedings of EvoWorkshops 2007. LNCS, vol. 4448, pp. 32–41 (2007).

[19] Biswas, A., Dasgupta, S., Das, S., Abraham, A.: Synergy of PSO and bacterial foraging optimization: a comparative study on numerical benchmarks. In: Proceedings 2nd International Symposium Hybrid Artificial Intelligent Systems (HAIS). Advances Soft Computing Series, Innovations in Hybrid Intelligent Systems. ASC, vol. 44, pp. 255–263. Springer, Germany (2007).

[20] Korani, W.: Bacterial foraging oriented by particle swarm optimization strategy for PID tuning. In: GECCO'08 Proceedings of the Genetic and Evolutionary Computation Conference. ACM, pp. 1823–1826. Atlanta (2008).

[21] Dasgupta, S., Biswas, A., Das, S., Panigrahi, B.K., Abraham, A.: A Micro-Bacterial Foraging Algorithm for High-Dimensional Optimization (2009).

[22] Chen, H., Zhu, Y., Hu, K.: Cooperative bacterial foraging optimization. Discret. Dyn. Nat. Soc. 2009.

[23] Dasgupta, S., Das, S., Abraham, A., Biswas, A.: Adaptive computational chemotaxis in bacterial foraging optimization: an analysis. IEEE Trans. Evolut. Comput. 13(4), 919–419(2009).

[24] Chen, H., Zhu, Y., Hu, K.: Multi-colony bacteria foraging optimization with cell-to-cell communication for RFID network planning. Appl. Soft Comput. 10, 539–47 (2010).

[25] Kim, D.H.: Hybrid GA-BF based intelligent PID controller tuning for AVR system. Appl. Soft Comput. 11, 11–22 (2011).

[26] Gollapudi, S.V.R.S., Pattnaika, S.S., Bajpaib, O.P., Devi, S., Bakwad, K.M.: Velocity modulated bacterial foraging optimization technique (VMBFO). Appl. Soft Comput. 11, 154–65 (2011).

[27] Okaeme, N.A., Zanchetta, P.: Hybrid bacterial foraging optimization strategy for automated experimental control design in electrical drives. IEEE Trans. Ind. Inf. 9, 668–8 (2013).

[28] Abd-Elazim, S.M., Ali, E.S.: A hybrid particle swarm optimization and bacterial foraging for optimal power system stabilizers design. Electr. Power Energy Syst. 46, 334–41 (2013).

[29] Mandeep Kaur ,Sanjay Kadam: A novel multi-objective bacteria foraging optimization algorithm (MOBFOA) for multi-objective scheduling, Applied Soft Computing , Volume 66, May 2018, Pages 183-195.

[30] Lv, X.; Chen, H.; Zhang, Q.; Li, X.; Huang, H.; Wang, G. An Improved Bacterial-Foraging Optimization-Based Machine Learning Framework for Predicting the Severity of Somatization Disorder. Algorithms 2018, 11, 17.

[31] Huang Chen, Lide Wang,Jun Di, and Shen Ping,: Bacterial Foraging Optimization Based on Self-Adaptive Chemotaxis Strategy, Computational Intelligence and Neuroscience, Volume 2020 | Article ID 2630104.

[32] Yufang Dan, Jianwen Tao; Knowledge worker scheduling optimization model based on bacterial foraging algorithm, Future Generation Computer Systems, Volume 124,2021,Pages 330-337,ISSN 0167-739X.

[33] Bo Yang, Xuelin Huang, Weizheng Cheng, Tao Huang, Xu Li, Discrete bacterial foraging optimization for community detection in networks, Future Generation Computer Systems, Volume 128,2022,Pages 192-204, ISSN 0167-739X.

[34] Sandeep Gogula, V. S. Vakula, Optimization for position and rating of distributed generating units using bacteria foraging algorithm to reduce power losses, International Journal of Cognitive Computing in Engineering ,Volume 4,2023,Pages 287-300,ISSN 2666-3074,

[35] C. Blake, E. Keough and C. J. Merz, UCI repository of machine learning database (1998). http://www.ics.uci.edu/~mlearn/MLrepository.html.