Journal of Engineering and Applied Sciences 12 (Special Issue 9): 8531-8536, 2017
ISSN: 1816-949X

A Conceptual Framework of Bacterial Foraging Optimization Algorithm for Data Classification

M. Hossin and F. Mohd Suria
Faculty of Computer Science and Information Technology (FCSIT),
Universiti Malaysia Sarawak (UNIMAS), 94300 Kota Samarahan, Sarawak, Malaysia

Abstract: Most previous researches on Bacterial Foraging Optimization Algorithm (BFOA) for data classification were integrated BFOA as a feature selection algorithm and parameters optimizer for other classifiers. To the best of our knowledge, no effort has been carried out to fully utilize BFOA as a classifier. This study presents a conceptual framework of instance-based BFOA. The proposed conceptual framework is designed based on the prototype searching approach whose target is to obtain an optimal reference set (cardinality) and simultaneously aim for high generalization performance by utilizing the strengths of BFOA.

Key words: Bacteria Foraging Optimization Algorithm, data classification, k-nearest neighbor, prototype selection, target, framework

INTRODUCTION

Many efforts have been done to invent and design more advanced classification algorithms at both data and algorithmic levels. These classification algorithms can be divided into five major categories which are neural network based, statistical based, rule based, instance selection and decision tree. This study will focus on Instance Selection (IS) approach for later discussion.

Historically, many IS algorithms have been employed to improve the performance of Nearest Neighbor (NN). The problem with NN classifier is that it requires a large storage of instances and long response classification time due to large storage (Garcia *et al*., 2012).

Therefore, the aim of IS algorithms is to reduce the training data as much as possible (or also known as sampling) and simultaneously attempt to achieve the highest possible classification accuracy using 1-Nearest Neighbor (1-NN) when dealing with the unseen data (Garcia-Pedrajas *et al*., 2010).

There are three different approaches of IS algorithms: noise filter, condensation and prototype searching (Garcia *et al*., 2012). Specifically this study will focus on the Prototype Searching (PS) approach.

At present, there are more than 50 available PS algorithms (Garcia *et al*., 2012; Garcia-Pedrajas *et al*., 2010). Most of these PS algorithms are derived from nature-inspired algorithms such as Genetic Algorithm (GA) (Kuncheva and Bezdek, 1998; Suguna and Thanushkodi, 2010; Gil-Pita and Yao, 2007), Evolutionary Algorithm (EA) (Garcia-Pedrajas *et al*., 2010; Cano *et al*., 2005), Ant Colony Optimization (ACO) (Aouidate and Baba, 2012, 2013) and Particle Swarm Optimization (PSO) (Hu and Tan, 2016). All these algorithms preserved its originality and applied stochastic heuristic search to locate the solution (a reference set) by transforming this searching problem into optimization problem.

In this study, we are proposing instance-based Bacterial Foraging Optimization Algorithm (BFOA) for data classification using prototype searching approach. To the best of our knowledge, there have been no effort proposed to fully utilize BFOA as a PS algorithm. However, there were a number of efforts incorporating BFOA as a part of other classifiers which aims are to improve and increase the classification performance. For instance, BFOA has been employed for feature selection (Jakhar *et al*., 2011; Rani and Mangat, 2013; Kora and Kalva, 2015; Bensujin *et al*., 2016; Rani *et al*., 2016; Sindhu, 2016) and optimizing the parameters of other classifiers (Hadi *et al*., 2011; Varghese *et al*., 2012; Chakrabarty *et al*., 2012; Qiang *et al*., 2013; Putra and Kom, 2014; Chen *et al*., 2014; Kaur and Kaur, 2014; Pal *et al*., 2014). We found that with the assistance of BFOA, it is able to increase the generalization performance of other classifiers.

Interestingly, BFOA has been adopted and employed as a single clustering algorithm (Wan *et al*., 2012). It shows outperformed result as compared to other clustering algorithms. This finding demonstrates that it is possible to adopt and adapt BFOA as

Corresponding Author: M. Hossin, Faculty of Computer Science and Information Technology (FCSIT), Universiti Malaysia Sarawak (UNIMAS), 94300 Kota Samarahan, Sarawak, Malaysia

Table 1: Summary of BFOA works on data classification

Data classification problems	Researchers	Role of BFOA
Feature selection algorithm	Jakhar *et al*. (2011)	BFOA was used to further reduce the extracted features of Cambridge Online Research Laboratory (ORL) gray-scale face dataset that made by Discrete Cosine Transform (DCT) technique
	Rani and Mangat (2013)	BFOA was adapted to select relevant features in the Pima Indian Diabetes dataset to improve the generalization ability of feed-forward back propagation neural network classifier
	Kora and Kalva (2015)	A hybrid BFOA and PSO known as BFPSO has been employed for feature selection of ECG signals and these selected features have been used as the input for Levenberg-Marquadt Neural network classifier
	Bensujin *et al*. (2016)	The Frequency Dependant Adaptive chemotactic Bacterial Foraging Optimization Algorithm (FDABFOA) has been proposed to optimize and fine-tune the extracted features of ST Elevation Myocardial Infraction (STEMI) dataset
	Rani *et al*. (2016)	BFOA was utilized at the pre-processing stage for feature selection in ANN classifier to discard irrelevant features in lung image dataset for the ANN and SVM classifiers
	Sindhu (2016)	BFOA was employed to select a small subset of informative or relevant genes from thousands of genes of cancer expression dataset
	Pal *et al*. (2014)	Hybrid BFOA and Learning Automata algorithm have been employed to identify the optimal features subset from a given imagery Electroencephalography (EEG) based on Barin-Computer Interfacing (BCI) dataset
Optimizing parameters classifiers	Hadi *et al*. (2011)	BFOA was applied to optimize the weights and parameters of feed-forward neural network for of several benchmark datasets
	Varghese *et al*. (2012)	BFOA was used to tune the backpropagation neural network parameters for MRI images of Alzheimer disease dataset
	Chakrabarty *et al*. (2012)	Employed BFOA to optimize the kernel function of Support Vector Machine (SVM) for hyperspectral image classification
	Qiang and Ai-Min (2013)	BFOA has been applied to optimize the operation parameters of SVM regression model
	Putra and Kom (2014)	The BFOA has been employed to optimize the weights and bias parameter of backpropagation neural network to predict Forex Gold Index (XAUUSD)
	Chen *et al*. (2014)	The hybrid PSO and BFOA was introduced known as BFPSO to optimize Neural Fuzzy Classifier (NFC) for several benchmark datasets and skin colour detection problem
	Kaur and Kaur (2014)	BFOA has been applied to adjust the weight and parameter values of feed-forward neural network and cascade-forward neural network

a single classifier. With this motivation, this study intends to modify original BFOA as a prototype searching algorithm to enhance the performance of 1-NN (Table 1).

MATERIALS AND METHODS

The Bacterial Foraging Optimization Algorithm (BFOA): Basically, BFOA mimics the foraging behavior of *Escherichia coli (E. coli)* bacteria which is referring to the foraging strategy of bacteria swarms in multi-optional function optimization (Passino, 2002). The foraging strategy is a method of animals for locating, handling and ingesting their food. Bacteria search and obtain nutrients in a manner to maximize energy intake. By sending the signal, it enables an individual bacterium to communicate with others. Healthy bacteria will be reproduced and poor foraging bacteria will be eliminated. The bacteria will keep repeating these processes in their lifetime. Technically, BFOA consists of chemotaxis, reproduction and elimination-dispersal as its principal mechanisms (Passino, 2002; Das *et al*., 2009).

Chematoxis: This process imitates the movement process of biological *E. coli* through swimming and tumbling. Through these processes, each bacterium will move to another direction by tumbling to different direction and swimming with the same direction for the entire lifetime.

Reproduction: During the reproduction process, all bacteria will be sorted based on the objective function. The least healthy bacteria will be eliminated, where the good bacteria will be kept and split into two and placed at the same location. This is important because this algorithm needs to keep a constant population size of the swarm.

Elimination and dispersal: In biological *E. coli* lifetime, the bacteria lifetime is also affected by gradual or sudden changes in the local environment. This situation can cause the whole swarm of bacteria to die or disperse at different locations. To imitate this process, the BFOA will eliminate each bacterium using P Probability and a new location is randomly initialized over the search space. This will encourage the exploration of the unvisited search space regions.

Through the review, it is obvious to say that the ability of BFOA in finding global optimal solution and high convergence rate had encouraged researchers to further explore and manipulate the advantages of BFOA in various domains and applications. The detail of BFOA can be found in Das *et al*. (2009) for further reading. The next study reviews the application of BFOA in data classification.

BFOA in data classification: Interestingly, all works done on BFOA for data classification focused on integrating the BFOA with the existing classifiers in order to improve the classification accuracy. None of the works applied BFOA as an instance selection algorithm using prototype searching approach (Table 1).

From the reviews, most of BFOA works on data classification focusing on adopting or employing BFOA for feature selection (Jakhar *et al*., 2011; Rani and Mangat, 2013; Kora and Kalva, 2015; Bensujin *et al*., 2016; Rani *et al*., 2016; Sindhu, 2016) by discarding the irrelevant features and picking the relevant features. The fitness function in BFOA can also be changed accordingly to the nature of the problems (Jakhar *et al*., 2011).

In addition, BFOA has also been proposed to support the training process of neural-based classifiers. Few works (Hadi *et al*., 2011; Varghese *et al*., 2012; Chakrabarty *et al*., 2012; Qiang *et al*., 2013; Putra and Kom, 2014; Chen *et al*., 2014; Kaur and Kaur, 2014; Pal *et al*., 2014) employed BFOA to optimize the weight of neural network and other classifiers by searching the optimal learning parameters for different classification datasets.

We also found that the BFOA structure is dynamic because it can be applied at any stage of data classification processes (Jakhar *et al*., 2011; Rani and Mangat, 2013; Kora and Kalva, 2015; Bensujin *et al*., 2016; Rani *et al*., 2016; Sindhu, 2016; Hadi *et al*., 2011; Varghese *et al*., 2012; Chakrabarty *et al*., 2012; Qiang *et al*., 2013; Putra and Kom, 2014; Chen *et al*., 2014; Kaur and Kaur, 2014; Pal *et al*., 2014). The principal mechanisms of BFOA are also flexible and can be modified or changed (Bensjin *et al*., 2016). Interestingly, BFOA shows better performances as compared to other algorithms. This shows to us that BFOA has some credits and potential to be adopted and adapted for other domain applications. Table 1 summarized all the above mentioned works of BFOA in data classification.

RESULTS AND DISCUSSION

A conceptual framework: We can say that the idea to adopt BFOA as a prototype searching algorithm as shown in Fig. 1 is novel. However, the idea to adopt nature inspired algorithms such as GA (Kuncheva and Bezdek, 1998; Gil-Pita and Yao, 2007), EAs (Garcia-Pedrajas *et al*., 2010; Cano *et al*., 2005), ACO (Aouidate and Baba, 2012, 2013) and PSO (Hu and Tan, 2015) as a PS algorithm has been done before.

By employing PS approach, the ultimate challenge is to design a good prototype set that produced minimal

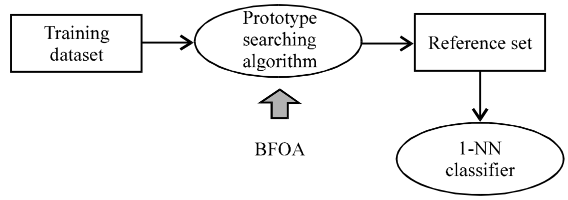

Fig. 1: IS-PS approach

reference set (cardinality) using BFOA and simultaneously to increase the classification accuracy of 1-NN. Another challenge is how to adapt the original principal mechanisms (chemotaxis, tumbling, swimming, reproduction and elimination-dispersal) of BFOA (Das *et al*., 2009) for prototype searching. The detail of this conversion process is described in the following subsections.

Data representation: There are two strategies that we can employ in order to design a good reference set either through selection or replacement (Suguna and Thanushkodi, 2010). For this particular conceptual framework of BFOA, the selection strategy has been chosen to represent a reference set. Through selection strategy, the reference set will be chosen from the existing data points by limiting the numbers. Let us assume that each bacterium B_{all} be the set of given j instances $B_{all} = \{x_1, x_2, ..., x_j\}$ where $j = 1, 2, ..., j$ are data from c classes in a particular dataset D. Let B be the selected data points and a reference set (solution), $B \epsilon B_{all}$. Every bacterium B is coded by binary string based on the length j and represented as 0 for inactive representative and 1 as an active representative.

Objective function: The most natural choice of objective function is to employ the classification accuracy. Classification accuracy is the ratio of correctly classified instances from x to overall number of instances j, $j = |x|$. However, in prototype searching method, instead of getting the highest classification, we are also targeting a minimal reference set (cardinality). Therefore, we proposed an objective function which combined the classification accuracy with penalty function (Kuncheva, 1997). The target is to find a set of B-prototypes B* that satisfies $B^* = \text{argmax}_{Scx} F(B)$ where F(B) is the objective function. The F is comprises two components:

$$F(B) = A(B) - \alpha f(|B|)$$

The first component A(B) denotes the classification accuracy when using B as the reference set. Meanwhile, the second component $\alpha f(|B|)$ denotes the function of cardinality of B weighted by the coefficient $\alpha>0$. Logically, the higher the cardinality, the higher the penalty. Since, the objective function is to obtain minimal cardinality, parameter T has been proposed. In this case, T is predefined value where it can force the BFOA to converge to predefined number of prototypes, $|B^*| = T$.

Population initialization: Let us assume that a set of N population is randomly generated and represented as $P = B_1, ..., B_N$. Each value for each cell $B_N = \{x_1, x_2, ..., x_j\}$ will be determined by the prespecified probability P_{init}. This parameter is important to control the generated selected representatives either to be sparse or dense (Kuncheva and Bezdek, 1998). Since, we proposed the objective function with penalty (Das *et al*., 2009), the recommended P_{init} is between the ranges of 0.9-0.95 which means that the search is starting with a set containing about 5-10% of data points. In other words, through P_{init}, its enabling the initialization process to control the number of selected reference set to be approximately approaching to T.

Chemotaxis: Similar to original BFOA, there are two parts in chemotaxis process which are tumbling and swimming. These processes mimic the exploitation concepts.

In tumbling process, we maintain the original concept of BFOA where tumbling process still occurs in random. However, the tumbling process of BFOA for data classification is a little bit different from the original tumbling process in BFOA. Instead of moving the bacteria itself randomly for data classification, we are moving randomly the selected representative in each bacterium using one step forward (+1) or one step backward (-1) approach. For the swimming process, the tumbling process is repeated using the same direction. This is similar to the concept of swimming in original BFOA. The swimming process is kept repeating if the performance of the current reference set for each bacterium is increasing. This process will stop if the performance of current reference set is lower than the previous one or reaches the maximum swimming iteration s.

Tumbling: The active (1) instance (ith) is selected randomly from j instances and move to its neighbor in random directions (one step forward (i+1) or backward (i-1)). The value of previous location will be changed (flipped) to 0 and activate its inactive neighbor instance as a new representative (change value from 0-1). If the neighbor instance is active, it will proceed with the following location which value is inactive. This process will produce a modified reference set. Importantly, the total number of reference set (cardinality) is unchanged. Then, the modified reference set is evaluated using objective function to determine the current performance.

Swimming: When the $F(B_{s+1})$ of a current modified reference set is better than the previous one $F(B_s)$, the modified reference set will be used for the next tumbling process using a similar direction. This swimming process will keep repeating and stops when a current reference set in B_{s+1} produce inferior result than B_s or the maximum iteration(s) of swimming counter is met.

Reproduction: For reproduction phase, the population of bacteria P is sorted according to the objective function. Healthiest bacteria (better fitness function) will be at the top and vice versa. Then, half of the bacteria population with poor performance will be eliminated from the population (Passino, 2002). Then, the remaining bacteria are duplicated to maintain the number of population N for the next iteration. The new population will undergo chemotaxis until maximum iteration of reproduction is met, N_{rp}.

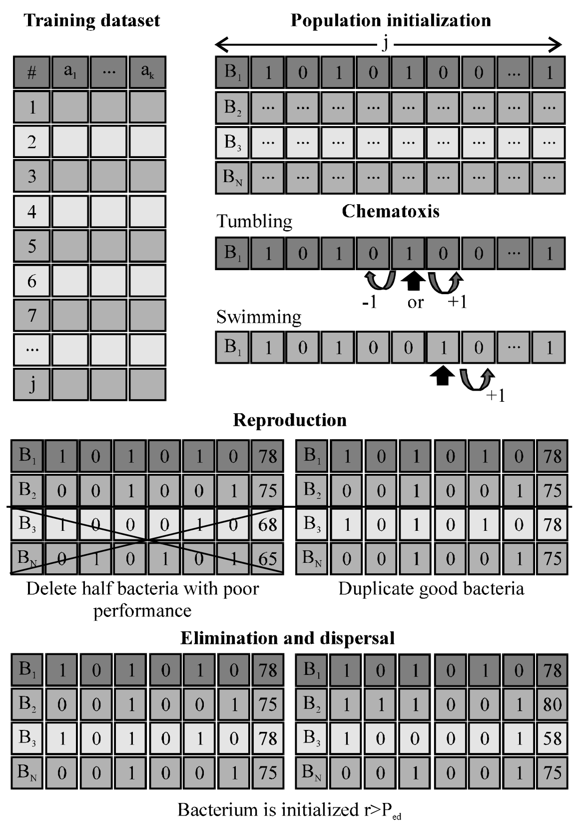

Fig. 2: BFOA framework using prototype searching approach

Elimination and dispersal: Bacteria in the population are reinitialized with new combination of instances with predefined probability, N_{ed}. Then, the chemotaxis and reproduction will repeat again until it triggers next iteration of elimination and dispersal, N_{ed}. The algorithm will stop until max iteration of elimination and dispersal is met. This process is crucial in facilitating BFOA to escape from local minima in order to find a global best solution which is similar to the concept in original BFOA in optimization. Higher value of iteration will make algorithm slower but more exploration in search space. The proposed framework of BFOA for data classification using prototype searching approach can be illustrated as shown in Fig. 2.

CONCLUSION

This study has presented a novel conceptual framework of BFOA for data classification using prototype searching approach. Technically, most parts of this proposed framework adapt the design of Genetic algorithm prototype based classifier. We believe this proposed framework can produce better classification accuracy by manipulating the global search capability of BFOA.

RECOMMENDATIONS

In the future, this proposed framework will be implemented and evaluated in terms of effectiveness and time efficiency and will be compared with other IS and PS algorithms.

ACKNOWLEDGEMENTS

This research was fully funded and sponsored by Ministry of Higher Education, Malaysia and Universiti Malaysia Sarawak, Malaysia under Fundamental Research Grant FRGS/ICT02(01)/1145/2014(12).

REFERENCES

Aouidate, M.A. and A.A.R. Baba, 2012. A Hybrid KNN-Ant Colony Optimization Algorithm for Prototype Selection. In: Neural Information Processing, Huang, T., Z. Zeng, C. Li and C.S. Leung (Eds.). Springer, Berlin, Germany, pp: 307-314.

Aouidate, M.A. and A.A.R. Baba, 2013. An efficient ant colony instance selection algorithm for KNN classification. Intl. J. Appl. Metah. Comput., 4: 47-64.

Bensujin, B., C.K.S. Vijila, J. Hemanth and C. Hubert, 2016. Frequency dependent adaptive chemotaxis in bacterial foraging optimization for ST segment elevation myocardial infarction prediction. Indian J. Sci. Technol., Vol. 9, 10.17485/ijst/2016/v9i1/71685.

Cano, J.R., F. Herrera and M. Lozano, 2005. Instance Selection using Evolutionary Algorithms: An Experimental Study. In: Advanced Techniques in Knowledge Discovery and Data Mining, Pal, R.N. and J. Lakhmi (Eds.). Springer, London, Englang, ISBN:978-1-85233-867-1, pp: 127-152.

Chakrabarty, A., O. Choudhury, P. Sarkar, A. Paul and D. Sarkar, 2012. Hyperspectral image classification incorporating bacterial foraging-optimized spectral weighting. Artif. Intell. Res., 1: 63-63.

Chen, C.H., M.T. Su, C.J. Lin and C.T. Lin, 2014. A hybrid of bacterial foraging optimization and particle swarm optimization for evolutionary neural fuzzy classifier. Intl. J. Fuzzy Syst., 16: 422-433.

Das, S., A. Biswas, S. Dasgupta and A. Abraham, 2009. Bacterial Foraging Optimization Algorithm: Theoretical Foundations, Analysis and Applications. In: Foundations of Computational Intelligence, Volume 3: Global Optimization, Abraham, A., A.E. Hassanien, P. Siarry and A. Engelbrecht (Eds.). Springer, Berlin, Germany, ISBN: 9783642010842, pp: 23-55.

Garcia, S., J. Derrac, J. Cano and F. Herrera, 2012. Prototype selection for nearest neighbor classification: Taxonomy and empirical study. IEEE. Trans. Pattern Anal. Mach. Intell., 34: 417-435.

Garcia-Pedrajas, N., J.A.R. del Castillo and D. Ortiz-Boyer, 2010. A cooperative coevolutionary algorithm for instance selection for instance-based learning. Mach. Learn., 78: 381-420.

Gil-Pita, R. and X. Yao, 2007. Using a Genetic Algorithm for Editing K-Nearest Neighbor Classifiers. In: Intelligent Data Engineering and Automated Learning, Yin, H., P. Tino, E. Corchado, W. Byrne and X. Yao (Eds.). Springer, Berlin, Germany, pp: 1141-1150.

Hadi, A.I.A.A., S.Z.M. Hashim and S.M.H. Shamsuddin, 2011. Bacterial foraging optimization algorithm for neural network learning enhancement. Proceedings of the 2011 11th International Conference on Hybrid Intelligent Systems (HIS), December 5-8, 2011, IEEE, Johor, Malaysia, ISBN:978-1-4577-2152-6, pp: 200-205.

Hu, W. and Y. Tan, 2016. Prototype generation using multiobjective particle swarm optimization for nearest neighbor classification. IEEE. Trans. Cybern., 46: 2719-2731.

Jakhar, R ., N. Kaur and R. Singh, 2011. Face recognition using bacteria foraging optimization based selected features. Intl. J. Adv. Comput. Sci. Appl., 1: 106-111.

Kaur, R. and B. Kaur, 2014. Artificial neural network learning enhancement using bacterial foraging optimization algorithm. Intl. J. Comput. Appl., 102: 27-33.

Kora, P. and S.R. Kalva, 2015. Hybrid bacterial foraging and particle swarm optimization for detecting bundle branch block. Springer Plus, 4: 1-19.

Kuncheva, L.I. and J.C. Bezdek, 1998. Nearst prototype classification: Clustering, genetic algorithms or random search?. IEEE Trans. Syst., Man, Cyber. Part C, 28: 160-164.

Kuncheva, L.I., 1997. Fitness functions in editing K-NN reference set by genetic algorithms. Pattern Recognit., 30: 1041-1049.

Pal, M., S. Bhattacharyya, S. Roy, A. Konar and D.N. Tibarewala *et al*., 2014. A bacterial foraging optimization and learning automata based feature selection for motor imagery EEG classification. Proceedings of the 2014 International Conference on Signal Processing and Communications (SPCOM), July 22-25, 2014, IEEE, Kolkata, India, ISBN: 978-1-4799-4665-5, pp: 1-5.

Passino, K.M., 2002. Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Syst., 22: 52-67.

Putra, I.K.G.D. and S. Kom, 2014. Optimized back propagation learning in neural networks with bacterial foraging optimization to predict forex gold index (XAUUSD). Appl. Math. Sci., 8: 37-40.

Qiang, S. ansd A.W.A.N.G. Min, 2013. A hybrid algorithm based on ls-svm and bacterial foraging approach for Burning-Through-Points (BTP) prediction in sintering process. J. Theor. Appl. Inf. Technol., 49: 231-240.

Rani, B., A.K. Goel and R.A. Kaur, 2016. Modified approach for lung cancer detection using bacterial foraging optimization algorithm. Intl. J. Sci. Res. Eng. Technol., 5: 39-42.

Rani, D. and V. Mangat, 2013. An efficient technique for disease diagnosis using bacterial foraging optimization and artificial neural network. Proceedings of the 2013 International Symposium on Computational and Business Intelligence (ISCBI), August 24-26, 2013, IEEE, Chandigarh, India, ISBN:978-0-7695-5066-4, pp: 100-104.

Sindhu, V., 2016. An efficient technique to cancer classification using fast improved bacterial foraging optimization. Intl. J. Recent Trends Eng. Res., 2: 436-441.

Suguna, N. and K. Thanushkodi, 2010. An improved k-nearest neighbor classification using genetic algorithm. Int. J. Comput. Sci. Issues, 7: 18-21.

Varghese, T., R.S. Kumari, P.S. Mathuranath and N.A. Singh, 2012. Performance Evaluation of Bacterial Foraging Optimization Algorithm for the Early Diagnosis and Tracking of Alzheimer's Disease. In: Swarm, Evolutionary and Memetic Computing, Panigrahi, B.K., S. Das, P.N. Suganthan and P.K. Nanda (Eds.). Springer, Berlin, Germany, pp: 41-48.

Wan, M., L. Li, J. Xiao, C. Wang and Y. Yang, 2012. Data clustering using bacterial foraging optimization. J. Intell. Inf. Syst., 38: 321-341.