Bio-inspired Algorithms for Diagnosis of Heart Disease

Moolchand Sharma, Ananya Bansal, Shubbham Gupta, Chirag Asija and Suman Deswal

Abstract Around one in every four deaths occur due to heart diseases in the USA every year (610,000 people approximately). One of the main reasons of fatality is due to a heart disease which also depends on various factors like obesity, diabetes, and aging. The deaths due to heart disease reduced by an indicative 41% in the USA between 1990 and 2016, whereas in our India it increased by approximately 34% from 155 to 209. The aim of this study is to aid the diagnosis of heart disease using bio-inspired algorithms. In this paper, a novel approach for the diagnosis of heart disease is inspected with the use of bio-inspired algorithms on Statlog (Heart) database from the UCI database. Bio-inspired algorithms used were binary ant colony optimization (ACO), binary firefly algorithm (FA), binary particle swarm optimization (PSO), and binary artificial bee colony (ABC) for feature selection. Bio-inspired algorithms target to decrease the dimensions of the dataset by defining the attributes which are most discerning. This helps us to successfully and efficiently classify whether a person is suffering from any heart disease or not. Out of the four algorithms, using the binary particle swarm optimization we have got the maximum accuracy of 90.09% and the classifier used was decision tree classifier. The results show that the algorithm is adequately quick and definite to be used in the analysis.

M. Sharma (✉) · A. Bansal · S. Gupta · C. Asija
Maharaja Agrasen Institute of Technology, Delhi, India
e-mail: moolchand@mait.ac.in

A. Bansal
e-mail: ananyabansal0@gmail.com

S. Gupta
e-mail: shubbham.gupta28@gmail.com

C. Asija
e-mail: chiragasija1@gmail.com

S. Deswal
Deenbandhu Chhotu Ram University of Science and Technology Murthal, Sonepat, (Haryana), India
e-mail: sumandeswal.cse@dcrustm.org

A. Khanna et al. (eds.), *International Conference on Innovative Computing and Communications*, Advances in Intelligent Systems and Computing 1087,
https://doi.org/10.1007/978-981-15-1286-5_45

Keywords Binary ant colony optimization (ACO) · Binary firefly algorithm (FA) · Binary particle swarm optimization (PSO) · Binary artificial bee colony (ABC) · Optimization · Bio-inspired · Feature selection

1 Introduction

The study demonstrates that deaths because of heart illnesses and strokes were progressively regular among the urban populace when the new century rolled over. Be that as it may, the trend has turned around from that point forward. Between the years 2000 and 2015, the age-institutionalized rate of mortality per 100,000 person-years has increased in rural men by over 40% and has declined men of urban states. For females, the expansion was over 56% in provincial (rural) India. The investigation centers around the age group of 30–69 years in both rural and urban India since deaths in this age groups are probably going to be preventable, and the post-mortem information for this age assemble will, in general, be progressively exacted. Death rates because of strokes declined in both rural and urban zones; however, the slide was more extreme in urban states of India. While coronary illness-related deaths have expanded in generally prosperous states, for example, Tamil Nadu, Karnataka, Punjab, and Haryana, stroke-related deaths have expanded in the relatively poorer states of the Northeast, West Bengal, and Chhattisgarh. While in the states of the Northeast, West Bengal, and Chhattisgarh, stroke death rates were around three multiple times higher than the average nationally. Deaths from heart-related malady among country Indians have surpassed the urban Indian people.

A few exertions have been made to automate the way toward classifying coronary illness by making predictions by using machine learning.

The efficiency of bio-inspired algorithms depends on balancing two main objectives: exploration (exploring the search space for promising solutions) and exploitation (exploiting best solutions found so far) [1].

Feature selection is the process of selecting out the most significant features from a full dataset in such course that subset investigates the target with an accuracy relatable to the efficiency of the original features set and with a less computational expense [2–4]. It is a fundamental and essential procedure in machine learning in modern times. Because of expansion in information in regard to features and instances, the quality of information has debased. The information is getting noisier which may prompt misleading and inaccurate results and misuse of memory. Subsequently, feature selection is important to expel this noisier data and get ideal outcomes. Feature selection is essentially partitioned into wrapper-based and filter-based algorithms. The factual methodology is utilized in the filter-based algorithm for scoring each element (feature) and wrapper-based algorithm costlier than the filter technique in regard to computational time and utilizes a machine learning algorithm for assessment [5, 6].

The key contributions of this work include:

- Four different algorithms are used for classification of the dataset: (i) K-nearest neighbors (KNN), (ii) decision tree, and (iii) random forest, and (iv) SVM.

- Random forest algorithm has shown results with 88.253% accuracy while decision tree has shown an increase of 1.84% from 88.253 to 90.093% using BPSO.
- Use of decision tree classifier reduces the computational time when compared with other classifiers.
- The experiment has been evaluated on the basis of accuracy, features selected, and complexity time.

2 Literature Review

Evolutionary algorithms are a processing algorithm which is influenced by and emulate natural biological processes so as to answer genuine real-life complex issues [7]. Numerous bio-inspired algorithms are being used, yet they all pursue a typical conviction which is to initialize a population of individuals and select the fittest among them which make the climb in the fitness of the population [8]. Other than from having a common belief, they indicate particular qualities that can be contemplated further profoundly into every method. According to the unified methodology [9] recommended by De Jong, the evolutionary algorithms must have these components, i.e., fitness function, parent's selection, mutation operator, crossover operator, survivor selection, representation, population, and termination condition. These segments portray the working of the algorithms and may vary from algorithm to algorithm. We have used the following four algorithms in our paper.

Particle swarm optimization [10, 11] depends on shared gathering nature of animals, for example, birds, fish schooling, and insect swarming, where the total target meets dependent on feedback from every single other unit of the group. Optimizing functions with broken (discontinuous), non-differential with a few non-linearly related parameters is the primary motivation behind this optimization [12].

Firefly algorithms [13–15] work far great than some other swarm optimization till present. It deals with the multi-modular goal to solve NP-hard situation with non-convex target functions. Population-based exploration is the premise with respect to which the firefly algorithm works; consequently, a few building fragments can be framed to get the required outcomes. The procedure keeps up equilibrium in search versus using as it gives better learning and understanding of parameter training. Firefly algorithm is additionally helpful as it may be used by different algorithms in hybridization to enhance results [16, 17].

Artificial bee colony (ABC) algorithm is a late proposed improvement methodology which reproduces the keen searching behavior of honey bees. A cluster of honey bees is called swarm which through social interest can successfully accomplish assignments given to them. In this ABC optimization, there are three sorts of honey bees: employed bees, onlooker bees, and scout bees.

The amount of employed bumblebees or the onlooker bumblebees is comparable to the number of solutions in the swarm.

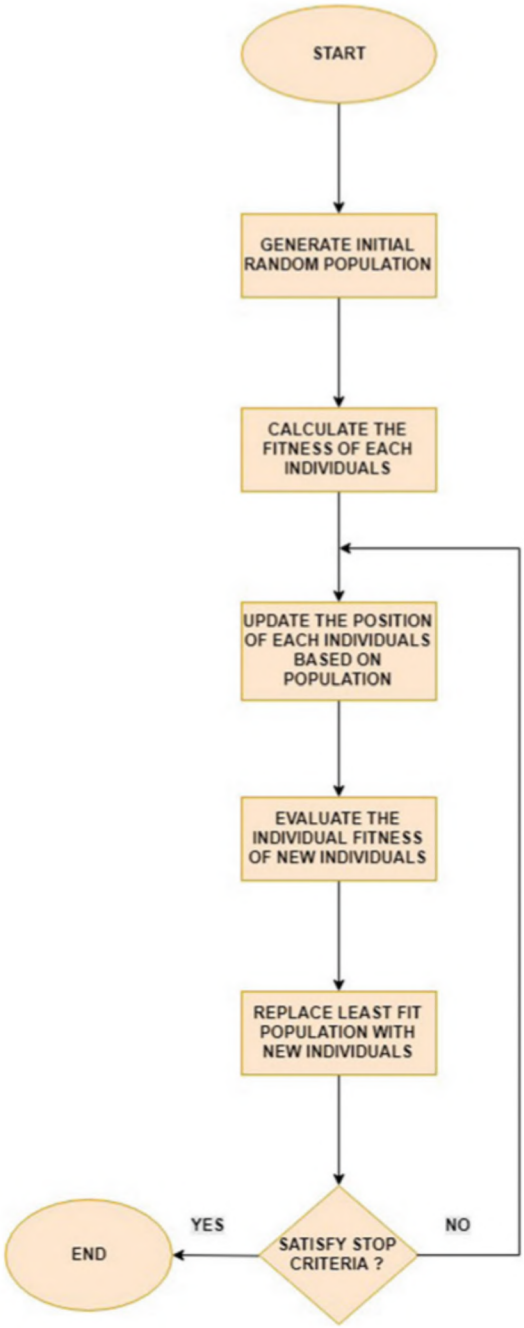

Fig. 1 Flowchart for general bio-inspired algorithm

The ant colony is a calculation for discovering perfect ways that rely upon the direction of ants chasing down sustenance or nourishment. Since the ant state works away at an incredibly one of a kind structure, the underground subterranean ant colony algorithm works amazingly well in charts with advancing topologies. Occurrences of such frameworks incorporate computer networks and artificial intelligence simulations of workers.

Below is the methodology used in this paper as shown in Fig. 1. This methodology is a general method to use any bio-inspired algorithm.

3 Proposed Work/Methodology

In this paper, we have initially performed feature selection by four bio-inspired algorithms. Each algorithm has been executed with four classifiers, i.e., decision tree (DT), random forest (RF), K-nearest neighbor (KNN), and support vector machine (SVM). Each algorithm has been executed for 20 times for better results.

Below is the accuracy which is shown by four algorithms (binary particle swarm optimization, binary ant colony optimization, binary artificial bee colony optimization, and binary firefly algorithm) as shown in Fig. 2.

3.1 Methodology

At first, we have executed dataset in four algorithms, i.e., binary ant colony optimization, binary artificial bee colony optimization, binary particle swarm optimization, and binary firefly algorithm. Firefly algorithm and particle swarm optimization have shown considerably superior results than other two algorithms. Figure 3 shows the number of attributes selected from above-mentioned algorithms.

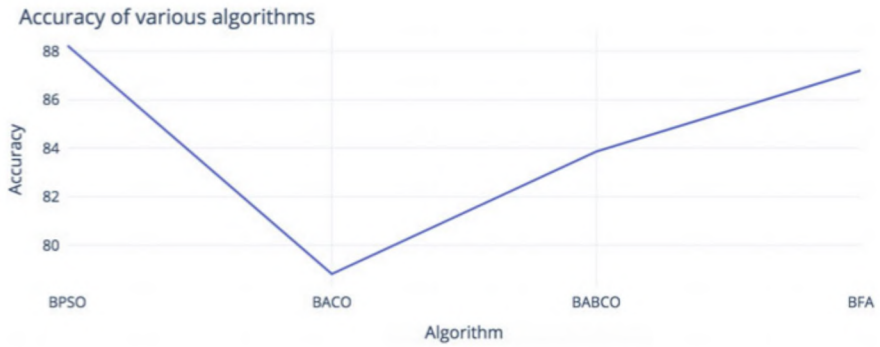

Fig. 2 Accuracy of the four algorithms used

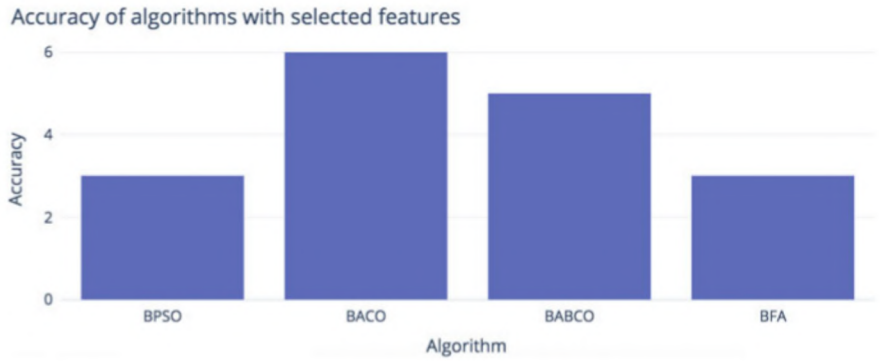

Fig. 3 Number of attributes selected from four algorithms

4 Implementation and Results

The algorithm has been run on numerous times with every one of the four classifiers. The experiment has been executed 20 times, and mean has been determined to improve results. By taking the mean, we can ensure that all cases are covered under our result.

4.1 Experimental Setup

The laptop used to execute the experiment has 5th Generation i5 core clocked at 2.6 GHz, with Ubuntu operating system and 8 GB of RAM with 128 GB Solid State Drive on MacOS High Sierra 10.13.5 operating system. Implementation and testing have been done on Python 3.6 using its libraries.

4.2 Input Parameters

4.2.1 Machine Learning Classifiers

For the data prediction, it is obligatory to make a data model classifier. Classification is a method of predicting class or label of provided data points. Any data type can be computed accordingly on classifiers. In this paper, we have used these four following classifiers:

- **Decision Tree**
  Decision tree classifier produces a lot of questions that are utilized to classify data together with its classes. Decision tree is easy to visualize and easier to

comprehend, and both numerical data and categorical data can be classified using it. The problem arises when a little variation is done which can change the entire tree and may prompt a completely separate tree which prompts high computational expense.

- **K-Nearest Neighbors**
  K-nearest neighbor method for classification is one of the least complex methods in machine learning and is an incredible method to acquaint yourself with machine learning and classification in general. At its most fundamental dimension, it is basically classifying by finding the most comparative data in the preparation data and making an educated guess based on their characterizations. Albeit extremely easy to comprehend and actualize, this technique has seen wide application in numerous spaces, for example, in recommendation systems, semantic searching, and anomaly detection.
- **Random Forest**
  Also known as random decision forests. It is the mainstream troupe strategy that can be utilized to construct predictive models for both classification and regression problems. Ensemble methods utilize different learning models to increase better prescient outcomes—on account of an r, the model random forest, makes as a whole forest of irregular uncorrelated decision trees to arrive at the most ideal answer.
- **Support Vector Machine**
  SVM is a strategy for preparing dataset in a hyperplane. In this, data is pointed in hyperplane and test data focuses are then contrasted and hyperplane points after which class is anticipated by the closeness of clusters shaped in the hyperplane. It is suggested for data with high dimensionality. The algorithm is memory effective and has the ostensible computational expense.

In Table 1, parameters for each classifier used in this experiment are shown.

Parameters essential for the execution of BPSO and BFA are shown in Tables 2 and 3, respectively.

Table 1 Input parameters for classifiers

Classifier	Parameter
Decision tree	Minimum depth = 30 Minimum sample split = 20 Maximum sample leaf = 1
K nearest neighbors	Neighbors = 3 Leaf size = 30
Random forest	Estimators = 25 Minimum sample split = 2
Linear support vector machine	Maximum iteration = 10

Table 2 Input parameters for BPSO

Parameters	Value
Number of population	50
Maximum iteration	200
Move rate	0.5
Limit search range	4

Table 3 Input parameters for BFA

Parameters	Value
Number of population	50
Maximum iteration	200
Alpha	0.25
Beta	0.5
Gamma	0.2

4.3 Dataset

The Statlog (Heart) dataset is used in this paper.

This dataset is a coronary illness database like a database present in the repository (heart disease databases) however in a marginally extraordinary frame. Two hundred and seventy records are present in the dataset with 13 attributes which are being used to foresee the heart illness. The scattered plot of a dataset is shown in Fig. 4.

Attribute Information:

1. age
2. sex
3. chest pain type (4 values)
4. serum cholesterol(mg/dl)
5. blood pressure resting
6. blood sugar fasting > 120 mg/dl
7. thal: 3 = normal; 6 = fixed defect; 7 = reversible defect
8. number of major vessels (0–3) colored by fluoroscopy
9. exercise-induced angina
10. maximum heart rate achieved
11. The slope of the peak exercise ST segment
12. old peak = ST depression induced by exercise relative to rest
13. resting electro-cardio graphic results (values 0, 1,2).

Types of Attributes:

Binary: 2, 6, 9
Nominal: 7, 3, 13
Ordered: 11

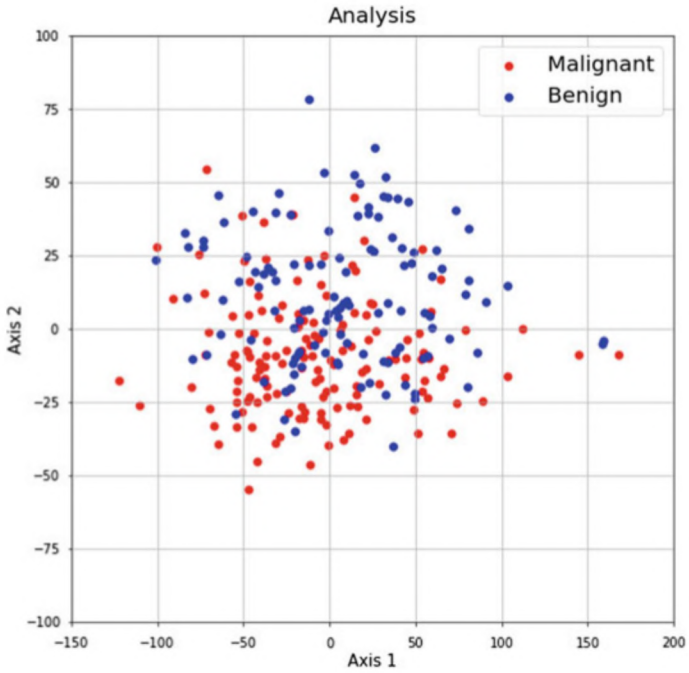

Fig. 4 Principal component analysis of dataset

Real: 1, 4, 5, 8, 10, 12.

Variable to be Predicted

Heart disease is absent (1) or present (2).

4.4 Results

Amid the experiment, the end condition has been set by maximum iteration as it can anticipate the maximum accuracy. During the underlying run, it is seen that the firefly algorithm has shown almost equal results to particle swarm optimization. The termination condition is not set by accuracy required as it might prompt an infinite loop as the algorithm is not capable of achieving that accuracy. What is more, it is hard to figure as far as possible as accuracy fluctuates with the classifier used.

Random forest algorithm has shown results with 88.253% accuracy while decision tree has shown an increase of 1.84% from 88.253 to 90.093% using BPSO. Decision tree has the least complexity time and has shown better results when used. Random forest had initial classification accuracies higher than that of decision tree, but decision tree had the advantage of the least time complexity. So, decision tree gave us better results when used with the bio-inspired algorithms.

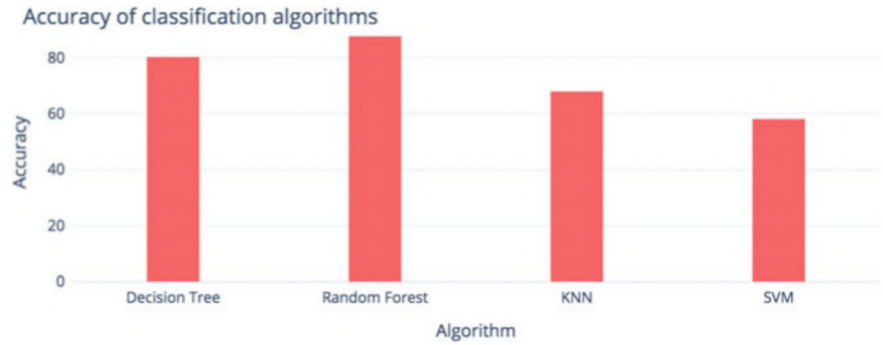

Fig. 5 Initial accuracy is given by machine learning classifiers

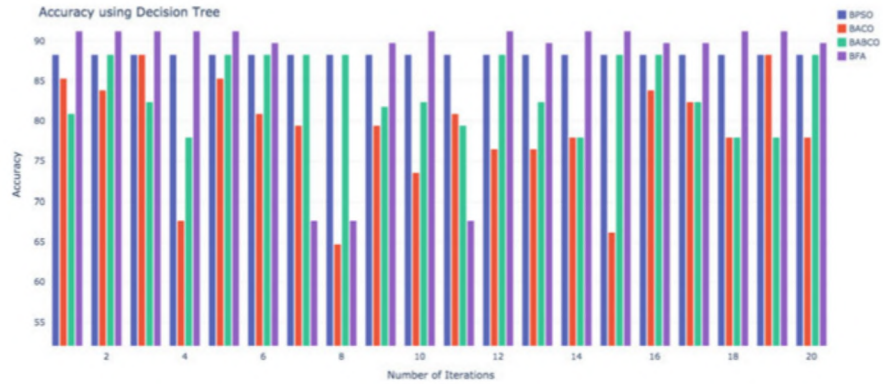

Fig. 6 Accuracy using DTC

BFA and BPSO performed at par when compared with all the four algorithms used, i.e., ant colony optimization, artificial bee colony optimization, particle swarm optimization, and firefly algorithm as shown in Figs. 5, 6, 7, and 8.

5 Conclusion and Future Scope

In this paper, the features are extracted using binary ant colony optimization (ACO), binary firefly algorithm (FA), binary particle swarm optimization (PSO), and binary artificial bee colony (ABC) along with four machine learning classifiers. Classifiers used are decision tree, K-nearest neighbor, SVM, and random forest classifiers. With decision tree classifier, maximum increase has been noted in the classification accuracy. So, we concluded that binary particle swarm optimization (PSO) along with decision tree classifier got us the best results.

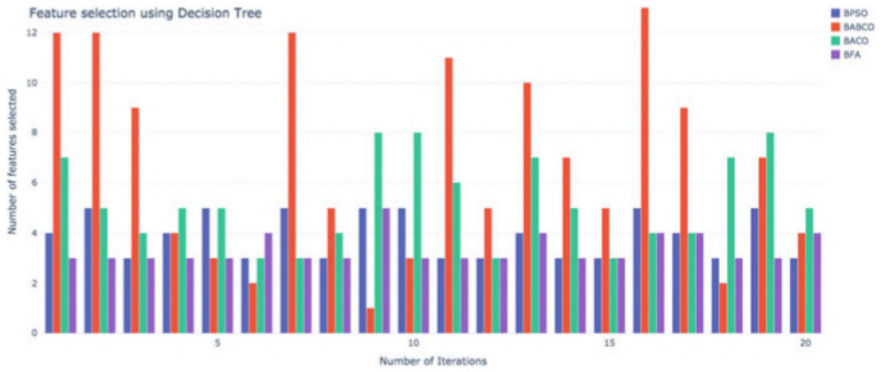

Fig. 7 Feature selection using DTC

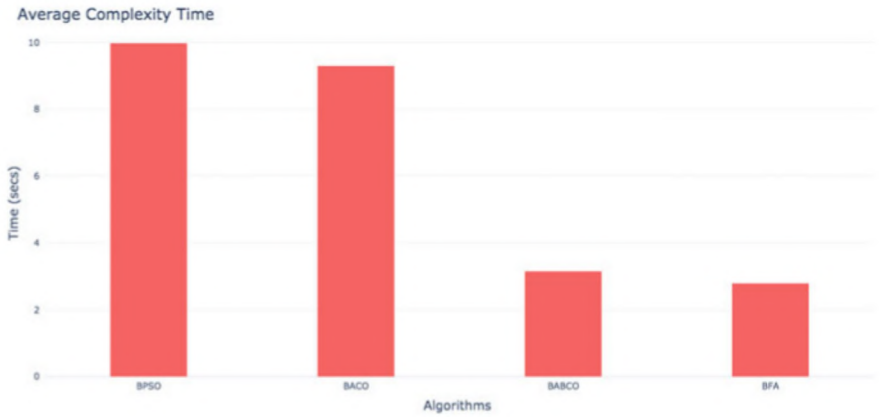

Fig. 8 Average time complexity

In future, hybridization of multiple algorithms can be used for better productivity of algorithms as they can give results with higher accuracy in equal or less computational time. After experimentation, we can report that after hybridization, there can be a rise in the accuracy of the predictions.

Highlights

- Decision tree, random forest, K-nearest neighbors, and support vector machine classifiers have been used.
- Computational time used by decision tree was least when compared to other classification algorithms.
- In the future, we could use hybridization for achieving better results.

References

1. G. Gopal, Hybridization in genetic algorithms. Int. J. Adv. Res. Comput. Sci. Softw. Eng. **3**, 403–409 (2013)
2. D. Gupta, J.J.P.C. Rodrigues, S. Sundaram, A. Khanna, V. Korotaev, V.H.C. Albuquerque, Usability feature extraction using modified crow search algorithm: a novel approach. Neural Comput. Appl. (2018). https://doi.org/10.1007/s00521-018-3688-6, SCIE (IF 4.2)
3. D. Gupta, A. Julka, S. Jain, T. Aggarwal, A. Khanna, V.H.C. de Albuquerque, Optimized cuttlefish algorithm for diagnosis of Parkinson's disease. Cogn. Syst. Res. **52**, 36–48 (2018) [SCIE (IF 1.4)]
4. D. Gupta, A. Ahlawat, Usability feature selection via MBBAT: a novel approach. J. Comput. Sci. **23**, 195–203 (2017) [SCIE (IF1.9)]
5. D. Gupta, S. Sundaram, A. Khanna, A.E. Hassanien, V.H.C. de Albuquerque, Improved diagnosis of Parkinson's disease based on optimized crow search algorithm. Comput. Electr. Eng. **68**, 412–424 (2018) [SCIE (1.57)]
6. R. Jain, D. Gupta, A. Khanna, Usability feature optimization using MWOA, in International conference on innovative computing and communication (ICICC) (2018)
7. X.-S. Yang, Nature-Inspired Metaheuristic Algorithms (Luniver Press, 2010)
8. A. Nayyar, S. Garg, D. Gupta, A. Khanna. Evolutionary computation—theory and algorithms, in *Advances in Swarm Intelligence for Optimizing Problems in Computer Science*, CRC Press
9. K.A. De Jong, *An Analysis of the Behavior of a Class of Genetic Adaptive Systems* (Doctoral dissertation, University of Michigan). Dissertation Abstracts International, 36(10), 5140B (University Microfilms No. 76-9381) (1975)
10. Y. Shi, R.C. Eberhart, Parameter selection in particle swarm optimization, in *Evolutionary Programming VII* (Springer Berlin, 1998), pp. 591–600
11. Y. Shi, R.C. Eberhart, An empirical study of particle swarm optimization, in *Proceedings of the 1999 IEEE Congress on Evolutionary Computation* (1999)
12. D. Floreano, C. Mattiussi, *Bio-inspired artificial intelligence: theories, methods, and technologies* (MIT Press, Cambridge, 2008)
13. A H. Gandomi, X.S. Yang, A.H. Alavi, Mixed variable structural optimization using firefly algorithm. Comput. Struct. (2011)
14. X.S. Yang, Firefly algorithms for multimodal optimization, in *Stochastic Algorithms: Foundations and Applications* (Springer Berlin Heidelberg, 2009), pp. 169–178)
15. X.S. Yang, S.S.S. Hosseini, A.H. Gandomi, Firefly algorithm for solving nonconvex economic dispatch problems with valve loading effect. Appl. Soft Comput. (2012)
16. A. Rahmani, S.A. MirHassani, A hybrid firefly-genetic algorithm for the capacitated facility location problem. Inf. Sci. (2014)
17. M.J. Ahammed, A. Swathi, D. Sanku, V. Chakravarthy, H. Ramesh, Performance of Firefly Algorithm for Null Positioning in Linear Arrays (2018), pp. 383–391. https://doi.org/10.1007/978-981-10-4280-5_40
18. M. Sahib, B. Ahmed, A new multi-objective performance criterion used in PID tuning optimization algorithms. J. Adv. Res. **115** (2015). https://doi.org/10.1016/j.jare.2015.03.004
19. T. Ojala, M. Pietikäinen, D. Harwood, A comparative study of texture measures with classification based on feature distributions. Pattern Recognit. **29**(1), 51–59 (1996)
20. D. Gupta, A. Khanna, Software usability datasets. Int. J. Pure Appl. Math. SCOPUS **117**(15), 1001–1014 (2017)
21. W.H. Wolberg, O.L. Mangasarian, Multisurface method of pattern separation for medical diagnosis applied to breast cytology, in Proceedings of the National Academy of Sciences, U.S.A., vol. 87 (1990), pp. 9193–9196
22. O.L. Mangasarian, R. Setiono, W.H. Wolberg, Pattern recognition via linear programming: theory and application to medical diagnosis, in *Large-scale numerical optimization*, ed. by T.F. Coleman, Yuying Li (SIAM Publications, Philadelphia, 1990), pp. 22–30