EasyChair Preprint

№ 1977

Cancer detection using Enhanced Powell's Bacterial Evolutionary based on k-means Algorithm

R Subhashini, Dhanush Reddy, Shiva Kumar, M Sai Deepu,
B Kanaka Naga Akhil, B Lalu Prasad Yadav,
K Anand Asish Kumar and M K Hema

November 18, 2019

Cancer detection using Enhanced Powell's Bacterial Evolutionary based on k-means Algorithm

Subhashini R[1]

Assistant Professor (Senior)
VIT University, Vellore, India

rsubhashini@vit.ac.in

C G Dhanush Reddy[2]

M.tech Integrated Software Engineering
VIT University, Vellore, India

cg.dhanushreddy2016@vitstudent.ac.in

Shiva Kumar[3]

M.tech Integrated Software Engineering
VIT University, Vellore, India

gandeshiva.kumar2016@vitstudent.ac.in

M Sai Deepu[4]

M.tech Integrated Software Engineering
VIT University, Vellore, India

msai.deepu2016@vitstudent.ac.in

P Kanaka Naga Akhil[5]

M.tech Integrated Software Engineering

VIT University, Vellore, india

kanaka.nagaakhil2016@vitstudent.ac.in

B Lalu Prasad Yadav[6]

M.tech Integrated Software Engineering

VIT University, Vellore, India

lalu.prasadyadav2016@vitstudent.ac.in

K Anand Asish Kumar[7]

M.tech Integrated Software Engineering

VIT University, Vellore, india

asishkumar2017@vitstudent.ac.in

M K Hema[8]

M.tech Integrated Software Engineering

VIT University, Vellore, India

mothukurikomala.hema2016@vitstudent.ac.in

1. Abstract:

Evolutionary algorithms frequently perform well approximating answers for a wide range of issues since they are in a perfect world don't make any supposition about the hidden wellness scene. Procedures from evolutionary algorithms connected to the demonstrating of organic development are commonly restricted to investigations of micro evolutionary procedures and arranging models

dependent on cell forms. Speaking the truth, this computational multifaceted nature is because of wellness work assessment. Be that as it may, apparently basic EA can fathom regularly complex problems along these lines, there might be no immediate connection between algorithm multifaceted nature and issue unpredictability.

2. Introduction:

The BEA (Bacterial Evolutionary Algorithm) draws motivation from a biological prodigy of microbial evolution. In difference to the ordinary mutation, hybrid and choice tasks in a GA (Genetic Algorithm), BEA combines two activities which are uncommon for developing its populace, to be specific the bacterial mutation and the quality exchange activity. An evolutionary algorithm (EA) is a subset of evolutionary calculation in artificial intelligence, a nonexclusive populace based metaheuristic enhancement algorithm. An evolutionary algorithm makes use of instruments stirred by natural advancement, for example, generation, transformation, recombination, and choice. Competitor answers for the streamlining issue assume the job of people in a populace, and the wellness capacity decides the nature of the arrangements.

As mentioned before, this algorithm several instruments which are stirred by natural advancement like recombination, generation, transformation and choice. These algorithms perform very well in approximating answers for a wide range of problems since all of them are in a perfect world without making any guess about the values that are hidden. It uses mechanisms inspired by nature and solves all processes by emulating the different behaviors of living organisms.

2.1. Uses and Applications:

- In some cases, it is extremely hard to find exact solution but sometimes a near optimal solution is sufficient. In these situations, evolutionary techniques can be effective. Because of their very random nature, evolutionary algorithms are certainly not assured to find an optimal solution for any problem, but will surely find a good solution if it exists.
- To set time table in schools and colleges. Since a teacher can teach in only one class at a time and the teacher can teach only in their area of expertise, it is a little difficult to thoroughly search for the optimal timetable.
- An Evolutionary Algorithm which is free of any human prejudices, can generate astonishing solutions that are comparable to, or better than, the best human-generated efforts. It is merely very much necessary that we recognize a good solution if at all it is presented to us, even if we don't know how to create a good solution. Putting it in other words, we need to be able to formulate an effective fitness function.

- A lot of engineers know pretty well about physics and who work in NASA. They know the right characteristics that make a good communications antenna. Designing an antenna is a little difficult. Even though the engineers know what is required from the final antenna, they may not know how to design the antenna which satisfies all those requirements.
- Evolvable Systems Group of NASA has made use of evolutionary algorithms to effectively progress antennas to make use of it on satellites. These progressed antennas possess irregular shapes with improper symmetry. It is not possible for a human to have arrived at such an unusual design. Despite this, when these antennas were tested, they proved to be extremely well adapted to their purpose.

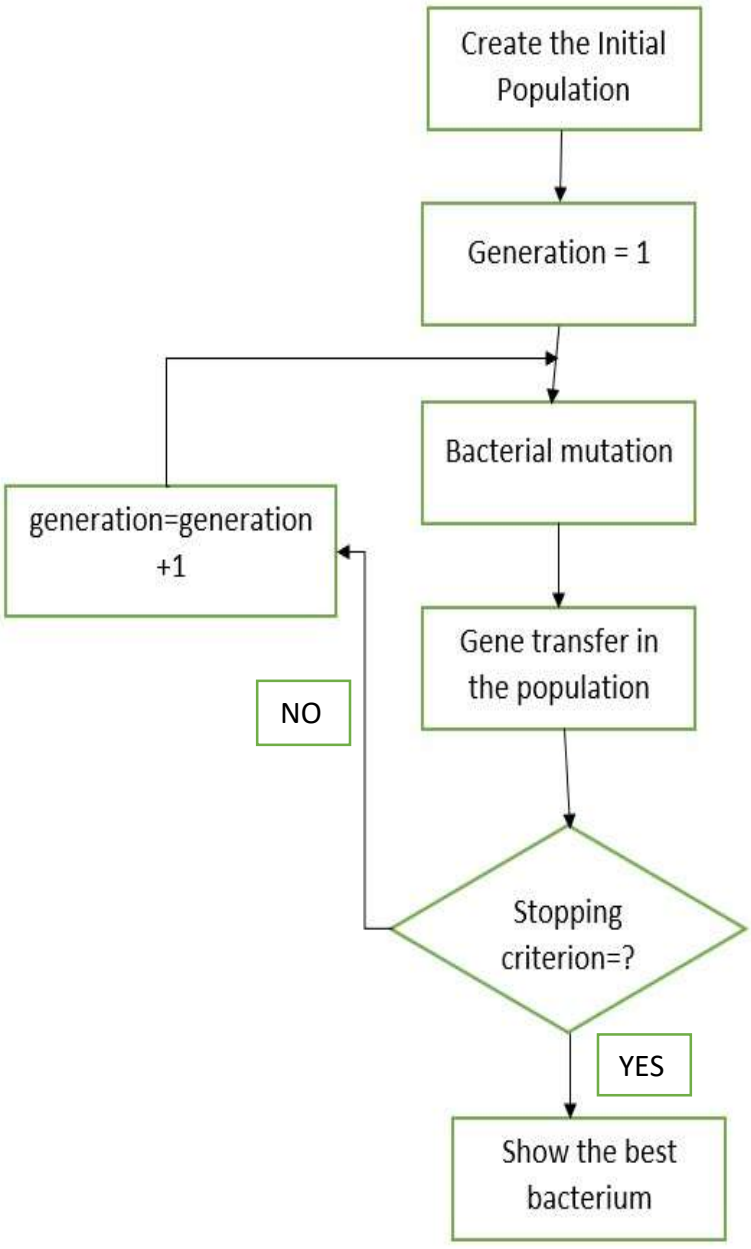
2.2. Basic evolutionary algorithm pseudocode:

Step-1: Create an initial population, it can be done in a random way.
*Step-2: Until it is done: (**exit criteria**)*
*Step-2.1: Select some pairs to be parents (**selection**)*
*Step-2.2: Combine pairs of parents to create offspring (**recombination**)*
*Step-2.3: Perform some mutation(s) on the offspring (**mutation**)*
*Step-2.4: Select some population members to be replaced by the new offspring (**replacement**)*
Step-3: Repeat

Firstly, an initial population has to be created. The creation of this population can be done in a random manner. Only after it is done the criteria has to be exited. Once it is done, select some pairs to be parents, this process is called selection. After the process of selection, combine the pairs of parents to create offsprings. This process is called recombination. Now the process of mutation has to be done. Perform some mutations on the offspring. Select some population members to be replaced by the new offspring. This process is called replacement. Repeat all the above mentioned steps. Best results can be acquired through more iterations. These steps have to be repeated again.

The creation of this population can be done in a random manner. Only after it is done the criteria has to be exited. Once it is done, select some pairs to be parents, this process is called selection. After the process of selection, combine the pairs of parents to create offsprings. This process is called recombination. Now the process of mutation has to be done. Perform some mutations on the offspring. Select some population members to be replaced by the new offspring. This process is called replacement.

2.3. **Flow Chart:**



2.3 Bacterial Evolutionary Algorithm

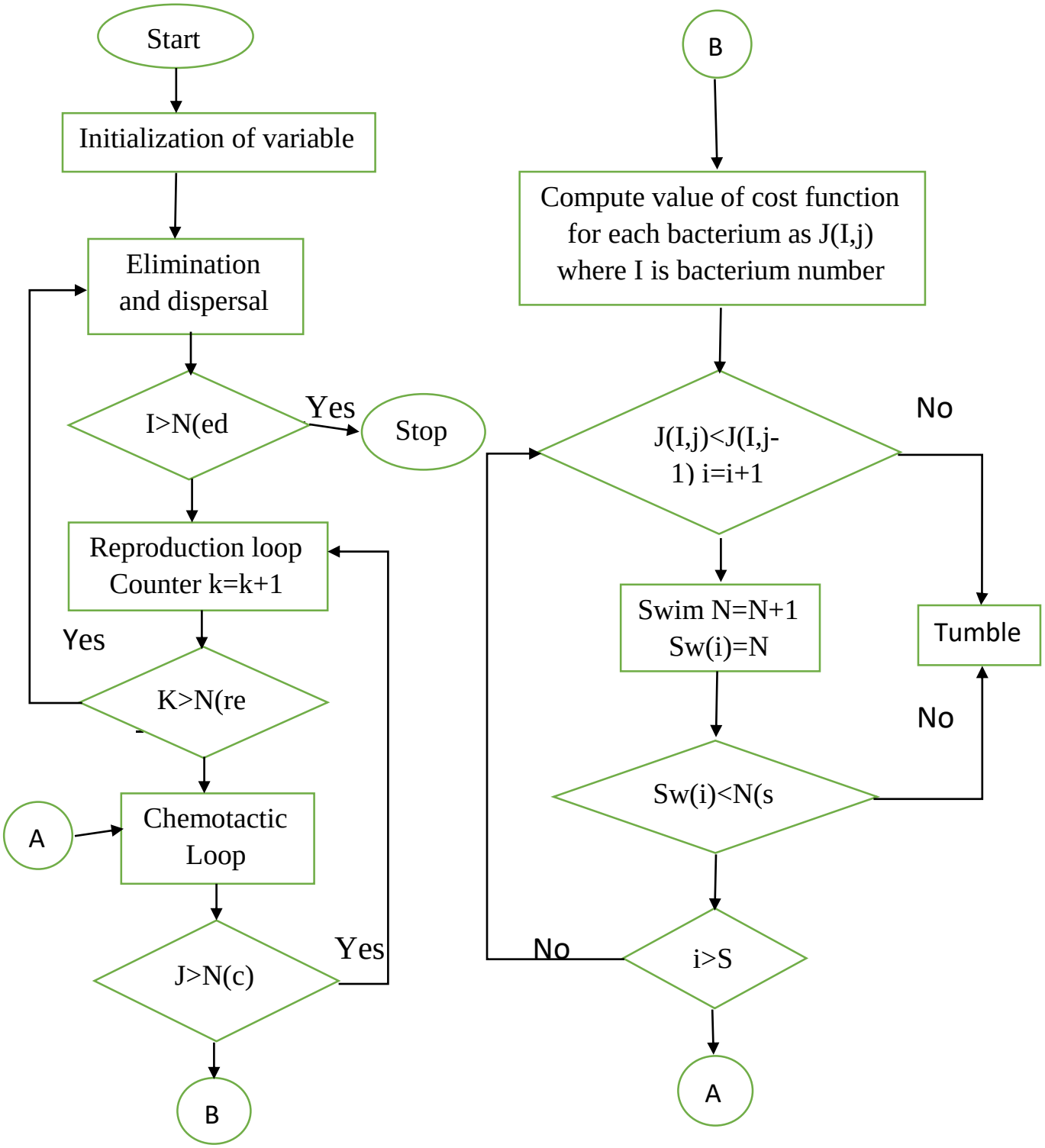
3. **Bacterial Foraging Algorithm:**

Bacteroids have their very own wellness work that mirrors the paces of vitality recharging and crash evasion. The normal for this calculation is that a determination of bacteroids is made by their condition with their passing as a trigger of the choice. This choice is, when all is said in done, done independent of the bacteroids claim wellness work. Irrespective of whether they have a higher wellness esteem at a given minute, they will kick the bucket when they are presented to an unexpected extreme ecological condition. On the off chance that a few bacteroids kick the bucket, the most grounded nearby bacteroids will assume control over their bodies by embedding their chromosomes, which ought to be most versatile in that neighborhood. Mutation is connected right now of this takeover to give the bacteroids a shot at development. The BEA is proper in a situation where numerous specialists like bacteroids are cooperating in one spot with numerous odds of connection.

3.1. Pseudocode:

Input: *Problem(size), Cells(num), N(ed), N(re), N(c), N(s), Step(size), d(attract), w(attract), h(repellant), w(repellant), P(ed)*
Output: *Cell(best)*
Step:1 - Population ← InitializePopulation(Cells(num), Problem(size))
Step:2 - For (l=0 to N(ed))
Step:3 - For (k=0 to N(re))
Step:4 - For (j=0 to N(c))
Step:5 - ChemotaxiAndSwim(Population, Problem(size), Cells(num), N(s), Step(size), d(attract), w(attract), h(repellant), w(repellant)
Step:6 - For (Cell belongs to Population)
Step:7 - If (Cost(Cell)<=Cost(Cell(best))
Step:8 - Cell(best) ← Cell
End
End
End
Step:9 - SortByCellHealth(Population)
Step:10 - Selected ← SelectByCellHealth(Population, Cells(num)/2)
Step:11 - Population ← Selected
End
Step:12 - For(Cell belongs to Population)
Step:13 - If(Rand() <= P(ed))
Step:14 - Cell ← CreateCellAtRandomLocation()
End
End
Step:15 - Return (Cell(best))

3.2.Flow chart:



3.2 Bacterial Foraging Algorithm

4. Data Set Information: (Breast Cancer detection)

This is one of three domains that has repeatedly appeared in the machine learning literature and provided by the Oncology Institute. This data set includes 201 instances of one class and 85 instances of another class. The instances are described by 9 attributes, some of which are linear and some are nominal.

4.1 Attribute Information:

1. Class: no-recurrence-events, recurrence-events
2. age: 10-19, 20-29, 30-39, 40-49, 50-59, 60-69, 70-79, 80-89, 90-99.
3. menopause: lt40, ge40, premeno.
4. tumor-size: 0-4, 5-9, 10-14, 15-19, 20-24, 25-29, 30-34, 35-39, 40-44, 45-49, 5054, 55-59.
5. inv-nodes: 0-2, 3-5, 6-8, 9-11, 12-14, 15-17, 18-20, 21-23, 24-26, 27-29, 30-32, 33-35, 36-39.
6. node-caps: yes, no.
7. deg-malig: 1, 2, 3.
8. breast: left, right.
9. breast-quad: left-up, left-low, right-up, right-low, central.
10. irradiat: yes, no

5. Data Preprocessing

The type of data Set Characteristics are multivariate. It has 286 instances. Area is based on life. Attribute characteristics are categorical. There are 9 attributes. Donated on 1988-07-11. Classification is one of the associated tasks. There are missing values. The number of web hits are 407463.

Here the steps followed are:

1. Import libraries
2. Read data
3. Checking for missing values
4. Checking for categorical data
5. Standardize the data
6. PCA transformation
7. Data splitting

Because there is so much correlation some machine learning models can fail.

5.1 Principal Component Analysis

We are going to create a Principal Component Analysis of the data. Principal Component Analysis (PCA) is a feature extraction method that uses orthogonal linear projections to capture the underlying variance of the data. Principal component analysis (PCA) is also a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables (entities each of which takes on various numerical values) into a set of values of linearly uncorrelated variables called principal components. PCA can be viewed as a special scoring method under the SVD algorithm. It produces projections that are scaled with the data variance. Projections of this type are sometimes preferable in feature extraction to the standard non-scaled SVD projections.

5.2. Classification using K-Nearest Neighbors:

Classifying breast cancer Using K-Nearest Neighbors:
Steps to compute K-NN algorithm:

1. *Determine parameter K = number of nearest neighbors.*
2. *Calculate the distance between the query-instance and all the training samples.*
3. *Sort the distance and determine nearest neighbors based on the K-th minimum distance.*
4. *Gather the category of the nearest neighbors*
5. *Use a simple majority of the category of nearest neighbors as the prediction value of the query.*

Building breast cancer classifier using K-NN algorithm:

The most important task in the healthcare field is the diagnosis of diseases. Many lives can be saved, if a disease is diagnosed early. Machine learning classification techniques can expressively benefit the medical field by providing an accurate and quick diagnosis of diseases. Hence, save time for both doctors and patients.

6. In classification section use bio-inspired algorithms

Genetic Algorithm

Genetic algorithm is an evolutionary based stochastic optimization algorithm with a global search potential. It follows the principle proposed by Charles Darwin which is Theory of survival of the fittest. Though, because of its outstanding performance in optimization, GA has been stared as a function optimizer. Algorithm begins by initializing a population of solution (chromosome). It includes representation of the problem usually in the form of a bit vector. Then for each chromosome evaluate the fitness using an appropriate fitness function suitable for the problem. Based on this, the best chromosomes are selected into the mating pool, where they undergo cross over and mutation thus giving new set of solutions(offspring). The below written pseudocode gives a clear explanation of how the algorithm works.

6.1. Pseudocode:

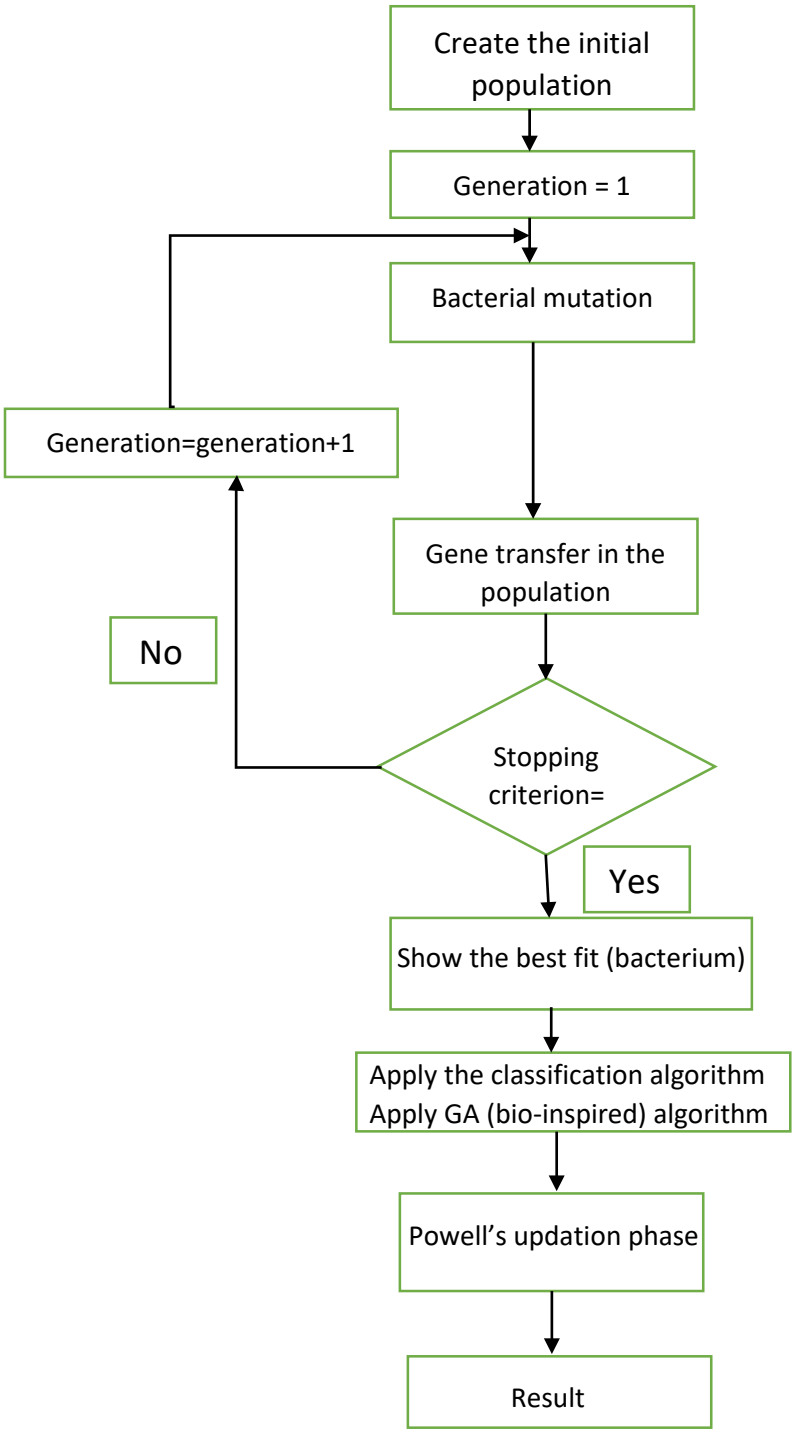
Input: *Population(size), Problem(size), P(crossover), P(mutation)*
Output: *S(best)*
Step:1- Population ← InitializePopulation(Population(size), Problem(size)
Step:2- EvaluatePopulation(Population)
Step:3- S(best) ← GetBestSolution(Population)
Step:4- While (¬StopCondition())
Step:5- Parents ← SelectParents(Population, Population(size))
Step:6- Children ← ø
Step:7- For(Parent 1, Parent 2 belongs to Parents)
Step:8- Child1, Child2 ← Crossover(Parent1, Parent2, P(Crossover))
Step:9- Children ← Mutate (Child1, P(mutation))
Step:10- Children ← Mutate (Child2, P(mutation))
End
Step:11- EvaluatePopulation(Children)
Step:12- S(best) ← GetBestSolution(Children)
Step:13- Population ← Replace(Population, Children)
End
Step:14- Return (S(best))

Powell's method, strictly Powell's conjugate direction method.

Powell's method, firmly Powell's conjugate direction method, is an algorithm which was projected by Michael J. D. Powell for finding a local minimum of a function. The function need not be differentiable, and no derivatives are taken. The function must be a real-valued function of a fixed number of real-valued inputs. The caller passes in the initial point. The caller also passes in a set of initial search vectors. Typically *N* search vectors are passed in which are simply the normal aligned to each axis. The method is generally used for calculating the local minimum of a continuous but complex function, especially one without an underlying mathematical definition, because it is not necessary to take derivatives. The basic algorithm is simple; the complexity is in the linear searches along the search vectors, which can be achieved via Brent's method.

Algorithm begins by initializing a population of solution (chromosome). It includes representation of the problem usually in the form of a bit vector. Then for each chromosome evaluate the fitness using an appropriate fitness function suitable for the problem. Based on this, the best chromosomes are selected into the mating pool, where they undergo cross over and mutation thus giving new set of solutions(offspring). Algorithm begins by initializing a population of solution (chromosome). It includes representation of the problem usually in the form of a bit vector.

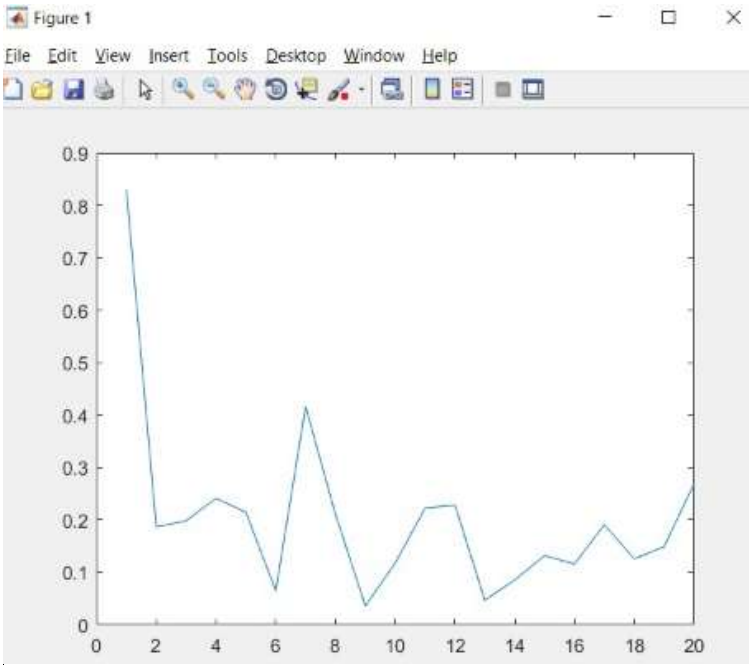
6.2. Flow chart:



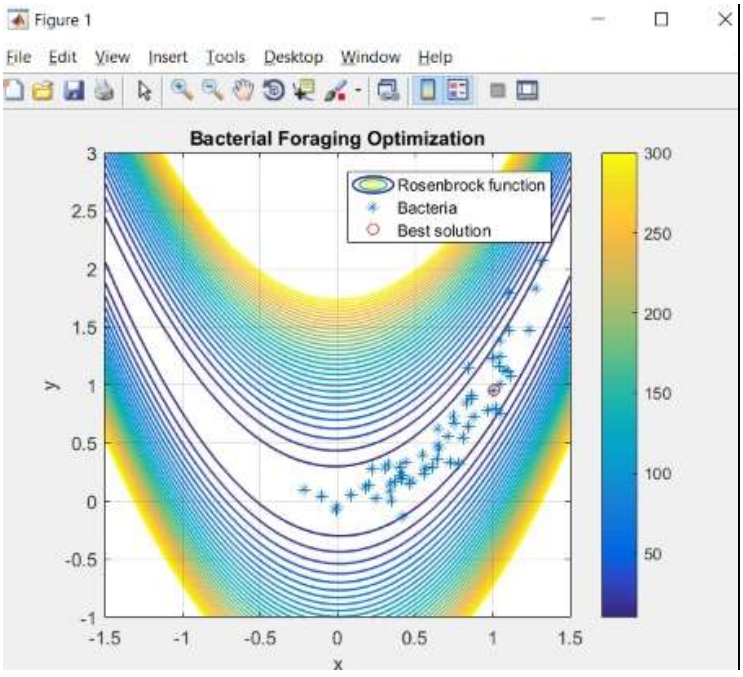
6.2 Updation phase using Powell's method

7. Conclusion:

Minimizing benchmark function which is called as Rosenbrock Function using Bacteria Foraging Optimization (BFO) technique.



Below output is for optimizing the two-variable Rosenbrock function using Bacterial Foraging Algorithm



8. References:

1. Das, S., Chowdhury, A., & Abraham, A. (2009, May). A bacterial evolutionary algorithm for automatic data clustering. In 2009 IEEE Congress on Evolutionary Computation (pp. 2403-2410). IEEE.
2. Gál, L., Lovassy, R., & Kóczy, L. T. (2015). Bacterial type algorithms used for fuzzy rule base extraction. Czasopismo Techniczne.
3. Das, S., Biswas, A., Dasgupta, S., & Abraham, A. (2009). Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications. In Foundations of Computational Intelligence Volume 3 (pp. 23-55). Springer, Berlin, Heidelberg.
4. Agrawal, V., Sharma, H., & Bansal, J. C. (2012). Bacterial foraging optimization: A survey. In Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2011) December 20-22, 2011 (pp. 227-242). Springer, India.
5. Jhankal, N. K., & Adhyaru, D. (2011, December). Bacterial foraging optimization algorithm: A derivative free technique. In 2011 Nirma University International Conference on Engineering (pp. 1-4). IEEE.