ISSN: 2229-6956 (ONLINE)
DOI: 10.21917/ijsc.2015.0147

ENHANCED BIO-INSPIRED ALGORITHM FOR CONSTRUCTING PHYLOGENETIC TREE

J. Jayapriya[1] and Michael Arock[2]
Department of Computer Applications, National Institute of Technology, Tiruchirappalli, India
E-mail: [1]jayajk2007@gmail.com, [2]michael@nitt.edu

Abstract

This paper illustrates an enhanced algorithm based on one of the swarm intelligence techniques for constructing the Phylogenetic tree (PT), which is used to study the relationship between species. The main scheme is to formulate a PT, an NP- complete problem through an evolutionary algorithm called Artificial Bee Colony (ABC). The tradeoff between the accuracy and the computational time taken for constructing the tree makes way for new variants of algorithms. A new variant of ABC algorithm is proposed to promote the convergence rate of general ABC algorithm through recommending a new formula for searching solution. In addition, a searching step has been included so that it constructs the tree faster with a nearly optimal solution. Experimental results are compared with the ABC algorithm, Genetic Algorithm and the state-of-the-art techniques like unweighted pair group method using arithmetic mean, Neighbour-joining and Relaxed Neighbor Joining. For results discussion, we used one of the standard dataset Treesilla. The results show that the Enhanced ABC (EABC) algorithm converges faster than others. The claim is supported by a distance metric called the Robinson-Foulds distance that finds the dissimilarity of the PT, constructed by different algorithms.

Keywords:

Phylogenetic Trees, Artificial Bee Colony Algorithm, Edit Distance, Converges Faster, Genetic Algorithm

1. INTRODUCTION

Phylogeny, multiple sequence alignment, motif prediction and genome analysis are some of the major research topics in sequence analysis [1]. Amongst them, multiple sequence alignment and phylogenetic analysis are used to answer many queries related to evolution and the relationships between a various pair of organisms. Phylogeny is the description of biological relationships of species, which can be represented as a Dendrogram tree. Evolutionary relationships are estimated by means of phylogenetic analysis. Based on the similarities or differences, living organisms are classified into groups and two closely related organisms are assumed to share a recent common ancestor [2].

To make predictions, concerning the tree of life is one of the significant uses of phylogenetic analysis of the sequences. To reconstruct evolutionary histories, there is a need of analysis of large numbers of taxa that requires fast but accurate algorithms [3]. Using the sequences of DNA/RNA/protein, the relationships of the organisms are studied. The main motivation for constructing the phylogenetic tree is to provide comprehend ancestry, and to understand the various evolving functions in different species for a biologist. This tree is also used to represents the most local conserved and globally conserved regions in a species. These regions are considered important for analyzing the different new genes.

Basically, constructing phylogenetic tree is considered as an NP-complete problem [4]. To solve these kinds of problems, we can use any bio-inspired algorithms. Within these algorithms, swarm intelligence has become a research interest to many scientists in related fields in recent years. The swarm intelligence is defined as the collective behavior of decentralized and self-organized swarms [5]. The intelligence of the swarm lies in the networks of interactions between simple agents and the environment. The examples of the swarm are bees swarming around their hive; an ant colony is a swarm of ants; a flock of birds is a swarm of birds and crowds is a swarm of people [6].

The contributions of this paper are:

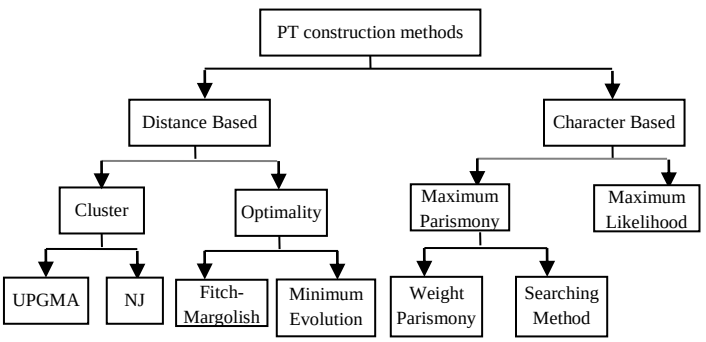
- proposing a variant of ABC algorithm
- introducing a new search equation used for exploitation
- and include a novel step for the exploration

to construct a phylogenetic tree.

This paper is organized as follows: The following section 2 presents the related work and section 3 gives the background for constructing a phylogenetic tree, the description of ABC algorithm, Genetic Algorithm (GA). In section 4, the proposed algorithm is described and in section 5 the experimental results are discussed. Eventually, the conclusions and future work of this paper are presented in section 6 conclusion.

2. RELATED WORK

The diagrammatic representation given in Fig.1 represents the different methods used for constructing PT. Two main categories of PT construction methods are character based and distance based that have its own merits and demerits. The first one is based on discrete characters of the sequence. The basic assumption is that the character at corresponding positions in sequence alignment is homologous among the sequences involved.



UPGMA – Unweighted Paired Group Method using Arithmetic average
NJ – Neighbor Joining

Fig.1. Phylogenetic tree and its construction methods

Under this category, there are two popular approaches namely the Maximum Parsimony (MP) and Maximum likelihood (ML) methods. The advantage of MP is intuitive. This approach tends to produce more accurate trees than distance based methods when the sequence divergence is low. But for high divergence, MP can be less effective. Considering the speed, MP is slower than distance methods. The second character based method ML uses probabilistic models to choose a good tree. When the number of taxa increases, it is impossible to use. To overcome this problem, several heuristic approaches are used. The second category is the distance based methods. This can be further divided into clustering or optimality based approaches. Depending on the optimality criteria, optimality based approaches classify into two types of algorithms, namely Fitch-Margoliash [7] and Minimum evolution. The main drawback of the optimality approach is slow when the large dataset is computed. In cluster-based methods, two different approaches are followed depending upon the assumptions. The first is the Unweighted Pair Group method using the Arithmetic average (UPGMA) [8] with the assumption that all the leaves (taxa) have equal distance from the root and the next is the Neighbor-Joining (NJ) [9] approach which assumes that the leaves have unequal distance from the root. Using the basic assumption, UPGMA starts grouping two taxa with smallest pairwise distance from the given distance matrix. This creates a reduced matrix by treating a new cluster as a single taxon. The grouping process is repeated until all taxa are placed on the tree. The last taxon added is the root of the tree constructed. Owing to its speed of calculation, UPGMA has found extensive use in clustering analysis. Many algorithms are there for constructing phylogenetic tree like clustering-based methods and heuristic based methods. Due to its increasing dataset, recent algorithms are based on heuristic approach. From the above study it is concluded that depending upon the alignment, different methods are chosen. When the amount of variation between the sequences is small, the maximum parsimony methods are chosen. When the variation is more or intermediate, the distance methods can be used. The maximum likelihood method is chosen when the sequences are more variable. The proposed algorithm uses the distance method, as it uses a large number of sequences with large variations.

In 2002, Ando et al., [10] developed an ant colony algorithm for constructing PT and compared the results with tool generated results. In 2004, Kumnorkaew et al., [11] proposed an ant colony optimization based algorithm for construction evolutionary tree. Lv et al., [12] in the year 2004 proposed the discrete PSO algorithm for PT that is not suitable for more than 40 sequences. In 2005, Perretto et al., [13] proposed a new phylogenetic tree construction method from a given set of objects (proteins, species, etc.). As an extension of ant colony optimization, this method proposes an adaptive phylogenetic clustering algorithm based on a digraph to find a tree structure that defines the ancestral relationships among the given objects. In 2006, Oh S. June et al., [14] proposed the kernel-based comparative analysis of metabolic networks for constructing phylogenetic trees. Evans et al., in 2006, [15] anticipated Relaxed Neighbor Joining (RNJ) algorithm for the construction of PT, in which the input distances of the dataset are purely additive. The main difference between NJ and RNJ is the computational time is less in latter than former. Qin et al., [16] in 2006 developed a new approach for the reconstruction of phylogenetic trees using ant colony optimization meta-heuristics. A tree is constructed using a fully connected graph and the problem is approached similarly to the well-known traveling salesman problem. In 2009, Katariya et al., [17] predict the DNA sequence in a PT using cellular automata which is not suitable for large datasets. Bhambri et al.,[18] in the year 2012 proposed the Kimura model for constructing PT.

3. BACKGROUND

PT is a two-dimensional graph showing evolutionary relationships among organisms. The following Fig.2 shows a basic tree. In this basic tree, the leaves represent genes species, which are termed as taxa. The internal nodes represent the hypothetical ancestral units. The root represents the common ancestor. The path from the root to a node is denoted as the evolutionary path between them. An example phylogenetic tree is shown in Fig.3. This figure shows the relationship of five species with root as carnivora, internal roots (internal nodes) as felidae, mustelidae, canidae and leaves nodes as panther, taxidea, lutra and canis. To construct a PT, a set of related sequences like DNA/RNA/protein are chosen. For phylogenetic analysis, this set of sequences should be reasonably aligned to find similarity/distance between them.

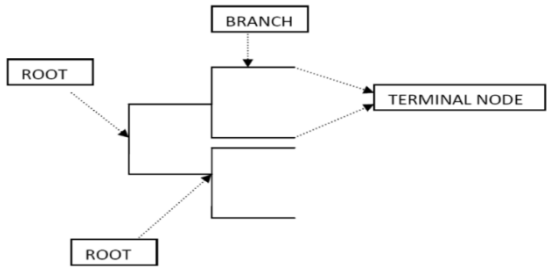


Fig.2. Simple Tree

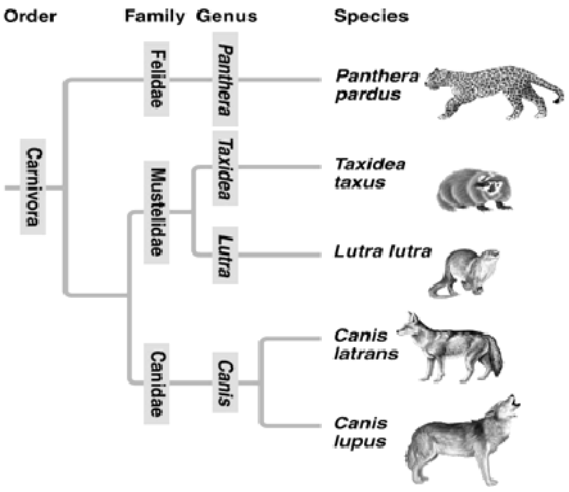


Fig.3. Example Phylogenetic tree (Courtesy: [19])

3.1 ABC OPTIMIZATION

Initially in 2005, Yang developed a Virtual Bee Colony (VBC) algorithm based upon the intelligent behavior of honey bee swarms [20] to solve the numerical optimization problems. The VBC algorithm has been introduced to optimize only the

function with two variables. For optimizing multivariable numerical functions, Dervis Karaboga proposed a bee swarm algorithm called ABC algorithm in 2005 as a technical report of numerical optimization problems [5]. This algorithm is based upon the intelligent foraging behavior of honey bees. And, it has gained wide popularity among researchers for its owing characteristics like few control parameters i.e., population size, limit and maximum cycle number [21]. It is more attractive than other optimization algorithms. It is simple, flexible, robust [22], converging faster and this can be easily hybridized with other optimization algorithms. For these reasons, the ABC algorithm is chosen and modified for constructing PT problem. In 2006, Karaboga and Basturk [23] developed the ABC algorithm shown in algorithm 1. The basic concept is the biological inspiration from the honey bees. The bees in a colony are divided into three groups, namely employed bees, onlooker bees and scouts. For each food source, there is only one employed bee. Therefore, the number of employed bees is equal to the number of food sources. The employed bees share information with the onlooker bees in a hive so that onlooker bees can choose a food source to forage.

Algorithm 1

General ABC Algorithm

Description:

1. Begin
2. Initialize the solution population, i = 1….SN
3. Evaluate population
4. Cycle = 1
5. REPEAT
6. Generate new solutions x_{mi} for the Employed bees using Eq.(1) and evaluate them using Eq.(3)
7. Keep the best solution between current and candidate
8. Select the visited solution for the onlooker bees by their fitness
9. Generate new solutions vmi for the onlooker bees using Eq.(2) depending upon the probability using Eq.(4). Then evaluate them using Eq.(3)
10. Keep the best solution between current and candidate
11. Determine if exists an abandoned food source and replace it using a scout bee
12. Save the best solution in memory obtained so far
13. Cycle = Cycle + 1
14. UNTIL cycle = maxcycle
15. End

$$x_{mi} = l_i + rand(0,1) * (u_i, l_i) \quad (1)$$

where, x_{mi} is input vector, u_i and l_i is upper and lower bound of the solution space, $rand(0,1)$ is the random number within the range 0 to 1. The second phase is the work of employed bee in which each of them goes to a food source to determine the neighbor food source using the Eq.(2),

$$v_{mi} = x_{mi} + \varphi_{mi}(x_{mi} - x_{ki}) \quad (2)$$

where, i is randomly selected parameter index, x_{ki} is randomly selected food source, φ_{mi} is a random number within the range -1 to 1. To find the global optimal, the fitness of food sources are essential. The fitness is calculated by the formula,

$$fit_i = \begin{cases} \frac{1}{1+f_i}, & if \ f_i \geq 0 \\ 1+abs(f_i), & if \ f_i < 0 \end{cases} \quad (3)$$

where, fit_i is fitness function, fi is objective function value. Next is the onlooker bee phase in the algorithm as the third one. Here higher food source is randomly selected depending upon the probability. The probability is calculated by the formula,

$$P_i = \frac{F(x_i)}{\sum_{j=1}^{S} F(x_j)} \quad (4)$$

where, S is number of food sources and $F(x_i)$ is fitness function. The onlooker bees search the neighborhood of food source using Eq.(2). Following the onlooker phase, the algorithm has scout bee's work as fourth and last phase. If there is no improvement in the food source for certain a trial, the solutions will be abandoned by scout bees. The new solution is generated using the expression Eq.(1).

Large numbers of real-world optimization problems have been solved by the ABC algorithm that demonstrates the utilization and effectiveness of this algorithm. The areas include Benchmark optimization, Bioinformatics field, Data Mining, Engineering designs and applications, Scheduling etc [24].

3.2 GENETIC ALGORITHM

GA was introduced by Holland in 1975 [25]. The search space of a problem is given as a collection of individuals in this algorithm. Each individual is referred to as chromosomes, which are represented by character strings. The main goal of GA is to converge fast and find the best individual genetic material from the search space. The quality of the individual is measured by the evaluation function. The Algorithm 2 shows the general algorithm for GA [26].

Algorithm 2

GA

Description:

1. Begin GA
2. Make initial population at random
3. REPEAT
   a. Select parents from the population
   b. Produce children from the selected parents
   c. Mutate the individuals
   d. Extend the population adding the children to it
   e. Reduce the extend population
4. UNTIL Certain conditions
5. End

According to the algorithm, the first step is to initialize the population at random for constructing PT. For PT construction problem, the combination of sequences is represented as path representation using permutation array as the first step. After measuring the evaluation function, the parents are selected using rank selection operator from the population and position

crossover operator is used to produce the children. Next step is to mutate the population, where the exchange mutation operator is used. As the last step, the population is reduced using the selection operation. Each iteration is referred as a generation.

4. PROPOSED WORK

4.1 DESCRIPTION

In order to apply the ABC algorithm effectively for constructing the phylogenetic tree, the EABC Algorithm 3 is proposed. Preprocessing task for constructing PT is to construct distance matrix, which is given as input for the proposed algorithm. The distance matrix is generated by calculating edit distance for the dataset (sequences). The minimum number of edit operations like insertions, deletions and substitutions between two strings are considered as the edit distance of that pair of strings [27].

Given two strings $S = s_i \ldots s_n$ and $S' = s_1', \ldots s_n'$, the edit distance can be calculated using the Eq.(5) given below:

$$D[i,j] = \min \begin{cases} ED[i-1,j]+1 \\ ED[i,j-1]+1 \\ ED[i-1,j-1]+0, if s_i = s_j'; \\ ED[i-1,j-1]+1, if s_i \neq s_j; \end{cases} \quad (5)$$

where, $ED[i,j]$ is the edit distance, i and j are indices of the sequences. The above method is one of the classical methods.

The most natural way of representing a set of sequences is with path representation using the permutation array. The combinations of the sequences in data are given as the path representation. The range of the path representation is between (1 to n) where n is the number of sequences. The Fig.4 shows the sample sequences and their permutation array. D1, D2, D3 and D4 are the distance between the respective sequences taken from the distance matrix.

(a) Sample sequence set

(b) Sample permutation array

Fig.4. Representation of sequences and permutation array

In this example, five sample sequences with the names as one to five are represented in Fig.4(a). The three different permutation arrays of the sequences are generated and distance for each pair like (3,4),(4,5)(5,2)(2,1) are summed, which is shown in Fig.4(b). The main aim is to select the best permutation arrays which have less in summation of the distances. In each sequence permutation array, for n number of sequences we have n-1 distances. Each path representation is defined as food sources and bees behave as an agent for generating new sources of food. Number of Employee Bees (EB) is initialized as $n/2$. The number of combinations generated should be equal to the EB which in turn equal to the Onlooker Bees (OB). The path representation which has the best fitness value in employee phase becomes as onlooker bee. The total population (food sources) is $FS = C/2$ where FS denotes the number of food sources and C denotes the colony size that is the sum of EB and OB. The different combinations of the trees are generated as total population using the expression Eq.(1). In this, u_i represents the number of sequences in the dataset and l_i represents one initial value.

Algorithm 3

EABC algorithm for constructing PT

Input: Distance matrix

Output: phylogenetic tree

Description:

1. Initialization
2. Generate population with different combinations of sequences in the dataset using random permutation Eq.(1)
3. Calculate the fitness value by using Eq.(8)
4. REPEAT
5. {Employed Bee Phase}
   a. for each employed bee
6. Call procedure EBphase
7. {Onlooker Bee phase}
   b. for each onlooker Bee
8. Call procedure OBphase
9. Memorize the best
10. {Scout phase}
11. Generate the new population randomly along with the best one using Eq.(6)
12. UNTIL some condition is met
13. Using the distances of the best combination of the sequences, the dendrogram tree is constructed

Algorithm 4

{Employee Bee Phase}

Procedure EBphase (POP)

1. Find the new population using Eq.(9)
2. Calculate the fitness value of Eq.(8)
3. IF (FF(new) > FF(old))
   a. for i = 1 to (EB/2)
      i. Call {procedure EBphase}
   b. end for
4. ELSE
   a. update with the new one
5. ENDIF
6. Return (POP)

Algorithm 5

{Onlooker Bee Phase}

Procedure OBphase(POP)

1. Find the new population using Eq.(10)
2. Calculate the fitness value of Eq.(8)
3. IF (FF(new) > FF(old))
   a. for i = 1 to (OB/2)
      i. Call {procedure OBphase}
   b. end for
4. ELSE
   a. update with the new one
5. ENDIF
6. Return (POP)

Using the distance matrix, the edge value between each pair in the population is calculated and this is considered as the fitness function. This can be expressed as Eq.(6):

$$e_i = Dx_i - Dx_j \tag{6}$$

where, e_i is edge value, Dx_i is distance of xi sequence and Dx_j is the distance of x_j sequence. The main objective of the EABC algorithm is to find an ordering of n sequences using permutation array which have minimum edge values. This permutation array is converted into a tree. For each tree, the fitness function is evaluated using the Eq.(7).

The objective function of this proposed algorithm is

$$FF = \min\left(\sum_{i=1}^{n} e_i\right) \tag{7}$$

where, FF denotes the fitness function. The steps in the employee phase are given Algorithm 4 in employee phase, the employed bees moved onto their randomly generated tree (food sources) and using the following Eq.(8) a new solution is produced.

$$EB(i) = \begin{cases} x_m(i) - x_m(\max), & \text{if } x_m(i) \le x_m(\max) \text{and } x_m(i) > x_m(\min) \\ x_m(\max), & \text{if } x_m(i) = x_m(\min) \end{cases} \tag{8}$$

where, $EB(i)$ is the ithpermutation array in the newly generated population, $x_m(i)$ is the i^{th} value in the permutation array x at a particular combination m, $x_m(\max)$ is the maximum value in the array x and $xm(\min)$ is the minimum value in the array x. The fitness function is calculated using the Eq.(7). The main idea of this equation is to encourage the randomness in the permutation array. Due to this, the combinations are changed without losing the best possibility solution. If $FF((x(i))>FF(EB(i))$ then recursively calls itself (employed phase) until certain condition else updates with the new one. Here, the condition used is half of EB times the employed phase recursively called to expedite the convergence. To increase the search space, we introduced a new step in the algorithm both in EB phase and OB phase. Next comes the onlooker phase, where the onlooker bee randomly selects the employee bee and evaluates the permutation array using the Eq.(9) that produces a new solution.

$$OB(i) = \begin{cases} x_m(i) - x_m(\max), & \text{if } x_m(i) \le x_m(\max) \text{and } x_m(i) > x_m(\min) \\ x_m(\max), & \text{if } x_m(i) = x_m(\min) \end{cases} \tag{9}$$

The fitness function is calculated using the Eq.(7). If $FF(EB(i)) > FF(OB(i))$ then recursively calls itself (onlooker phase) until certain condition else updates with the new one. Here, the condition used is half of OB times the onlooker phase recursively called to expedite the convergence. These steps are summarized in Algorithm 5. Next step is to memorize the best solutions that are used in the next generation. The goal of this proposed algorithm is to minimize the fitness value. The lower a fitness value, the better the solution. The unfit employed bees change into scout bees, to generate the new population randomly along with the best of the onlooker bee using the expression, Eq.(1). The optimal solution that is selected for the tree construction is considered as the best combination of sequences. Using this best set, the dendrogram tree is built.

4.2 ALGORITHM COMPLEXITY

The Time Complexity (TC) for each phase is calculated and the maximum of that is concluded as TC of the proposed algorithm. In this paper, finding the distance between the sequences are considered as the preprocessing step. The TC of this preprocessing is $O(nm)$ where, n is the number of sequences of length m each. In proposed algorithm, the initialization is the first phase to generate the population for n sequences. So, the TC of this phase is given as $O(n)$. Next comes the employee bee phase, in which it has an inner loop depending upon the condition. Here, two cases can be considered: one is the worst case in which it takes $O(n/2)$ and another is the best case in which it takes $O(n)$. Therefore, in the worst case it is $O(n^2)$ and in the best case it is $O(n)$. For onlooker phase also the same TC is needed for both cases. Next is the scout phase which takes $O(n)$. For constructing dendrogram tree, it takes $O(nm)$. Depending on the relationship between n and m, it can be written as $O(m^2)$ and this is considered as maximum. The number of generations is decided depending upon the number of data (sequences) taken. It can be said that k is constant which represents the generations (outer loop of the algorithm). Based upon the number of sequences n, k is decided. It is concluded that the TC of the proposed algorithm is given as $O(km^3)$.

5. EXPERIMENTAL RESULTS

In this section, the results are analyzed using two different datasets. Experimental parameters used by the algorithm are represented in Table.1. The algorithm is implemented with 16GM RAM, Windows 7 and MATLAB 13a in core i7 processor environment. The first dataset with seven sequences (D1) is cDNA sequences p53 whose description is given in Table.1. More details about this data can be found and download from the website [29]. First the preprocessing of the algorithm is done ie., the Edit distance is calculated and shown in Fig.3. This figure shows the distance between all the possible combinations of the dataset D1. When the distance between any two sequences is less, then they are considered to have more conserved regions. The following Fig.4 shows the PT constructed using this distance by UPGMA(T1), NJ(T2), RNJ(T3) and EABC(T4)

algorithms for the dataset D1 where T1, T2, T3, T4 are the trees constructed by the techniques respectively. The online code has been used to take the results for the state-of-the-art techniques. This shows that the proposed algorithm builds a PT similar to that of the techniques.

Table.1. Experimental parameters

Parameters	Algorithms		
	GA	ABC	EABC
No. of generations	200	200	200
No. of populations (no. of EB/OB)	n/2	n/2	n/2
Lower value	#	1	1
Upper value	#	n	n
Crossover	Position	#	#
Mutation	Exchange	#	#
Selection	Rank	#	#

- denotes not applicable

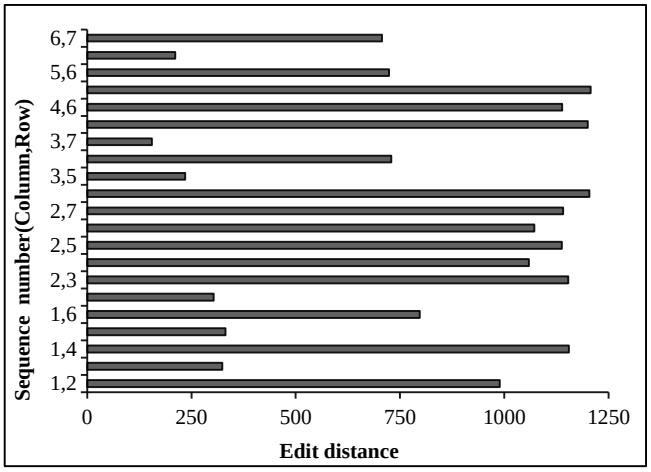


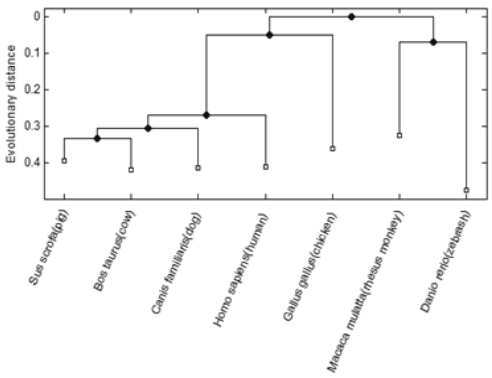
Fig.5. Edit distance

Table.2. Dataset D1

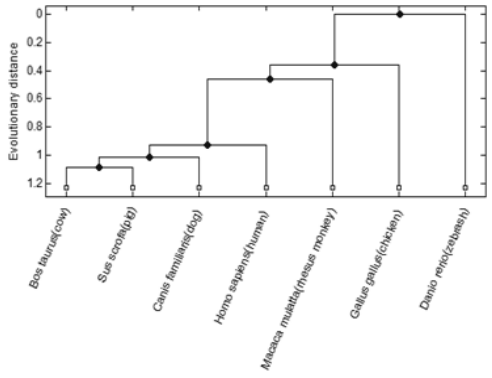
ID	Sequence length	Organism	Description
X02469	1317 bp	Homo sapiens (human)	mRNA for p53 cellular tumor antigen
L20442	2184 bp	Macacamulatta (rhesus monkey)	p53 mRNA, complete cds
X81704	1161 bp	Bostaurus (cow)	p53 Mrna
U60804	2105 bp	Daniorerio (zebra_sh)	tumor suppressor p53 (p53) mRNA
AF060514	1174 bp	Canisfamiliaris (dog)	p53 protein (p53) mRNA
X13057	1555 bp	Gallus gallus (chicken)	mRNA for nuclear oncoprotein p53
AF098067	1163 bp	Susscrofa (pig)	tumor suppressor p53 (p53) mRNA

Table.3. Efficiency comparison for ABC, GA and EABC algorithms with different datasets

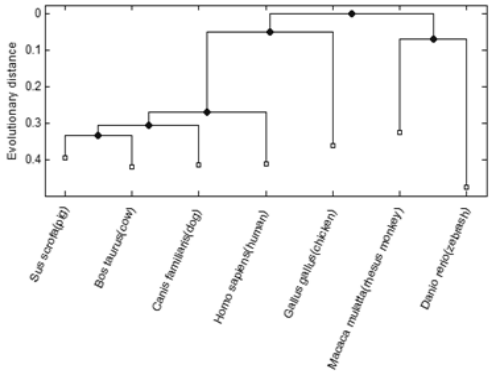
Dataset	Algorithm Type	Fitness Value			Time
		Avg.	Best	Worst	
D1 (7 sequences)	ABC	3475	3846.5	4218	0.70090
	GA	3712.5	3545	3796	0.483929
	EABC	3671	3462	3880	0.687839
D2 (15 sequences)	ABC	1978	1897	2059	0.689639
	GA	2207	2114	2300	0.458860
	EABC	2040	1860	2248	0.614124



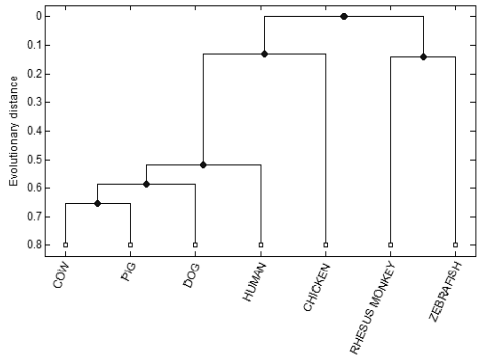
(a) NJ



(b) UPGMA



(c) RNJ



(d) EABC

Fig.6. PT constructed by NJ, UPGMA, RNJ, EABC for the dataset D1

Another dataset (D2) [29] is Treezilla which contains 500 seed-plants sequences of genes. Among this, 60 sequences are taken for the experiment. Fig.5 represents the convergence rate of the ABC, GA and EABC algorithms of data D1.

From Fig.7, it is clear that the time taken for the EABC algorithm is less than UPGMA, NJ, RNJ and ABC. Even though GA algorithm takes less time than other algorithms, it does not yield minimum fitness value that is shown in Table.3. This table shows the fitness function value for D1 of seven sequences and D2 of fifteen sequences. This clearly explains that the time taken by the proposed algorithm is less than the state-of-the-art techniques. The computation time depends on the number of sequences and its length. The result shows that all the algorithms have taken less time for D2 than D1. This infers that when the length of sequences decreased and the number of the sequences increased, the time taken by the algorithms is less. In this paper, the dataset D1 has seven sequences with an average length of 1522 bp(basepair) approximately and for the dataset D2 sixty sequences which have an average length of 1398 bp.

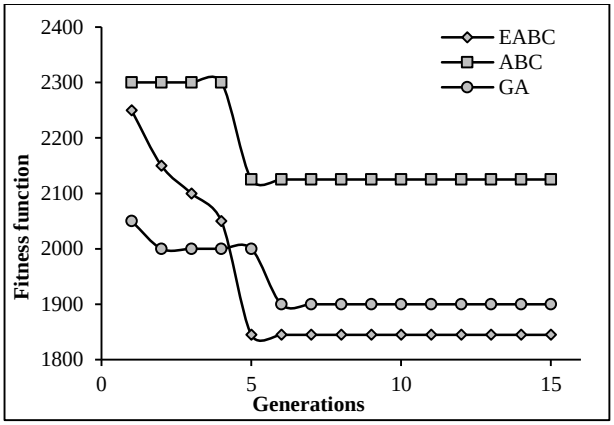


Fig.7. Convergence rates of ABC, GA and EABC algorithms for D1

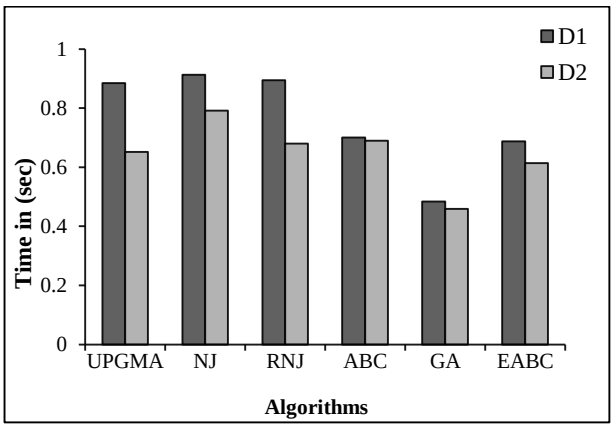
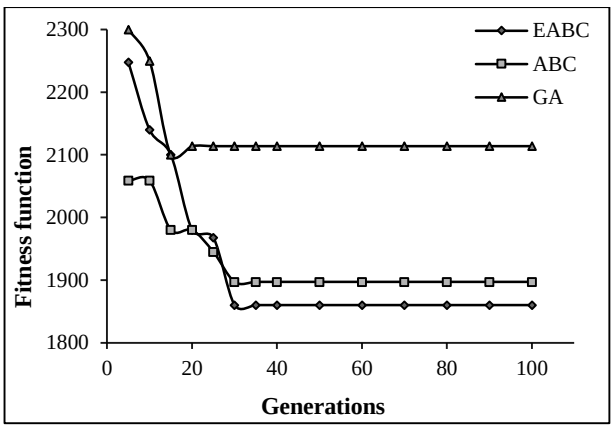
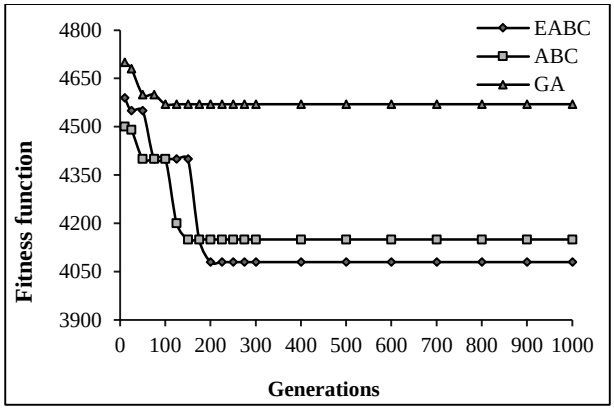


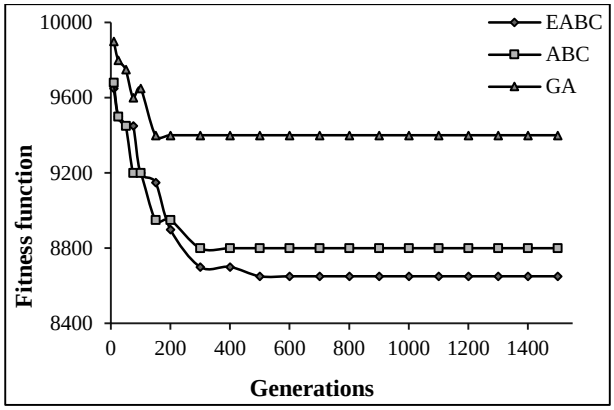
Fig.8. Time efficiency comparison of D1 & D2



(a) Fitness function convergence for D2 with 15 sequences



(b) Fitness function convergence for D2 with 30 sequences



(c) Fitness function convergence for D2 with 60 sequences

Fig.9. Convergence rate of ABC, GA and EABC algorithms for D2

As a result, when the length of the sequence and the number of the sequences are increased, the computational time also increased. The following Fig.9 represents the convergence of GA, ABC and EABC algorithms for the different number of sequences like 15, 30 and 60 respectively for the dataset D2. Initially, for the fewer number of sequences, the proposed algorithm converges faster than other above-mentioned algorithms and gives an optimal solution. This is shown in Fig.9(a) and (b). And, Fig.9(c) shows that the EABC algorithm converges slowly than ABC and GA algorithm, however, yields an optimal solution when the number of sequences is increased. However,the convergence time of GA is less; it doesn't yield less fitness value. These results conclude that the proposed EABC algorithm have the minimum fitness value in best and average cases. Therefore, it is concluded that the proposed algorithm yields the minimum fitness value in less time for constructing PT. The constructed PT using UPGMA, NJ, RNJ and EABC are compared with each other using Robinson–Foulds (RF) distance [30]. This distance gives the dissimilarity between the trees using the following Eq.(10).

$$d_{RF}(T1,T2)=\frac{|C(T1)\setminus C(T2)|+|C(T2)\setminus C(T1)|}{2} \qquad (10)$$

where, d_{RF} is the Robinson-Foulds distance, $C(T1)$ and $C(T2)$ are the cluster collections of the *PT*, *T*1 and *T*2 respectively. The following Fig.8 shows the dissimilarity between *T1*, *T*2 and *T*3.

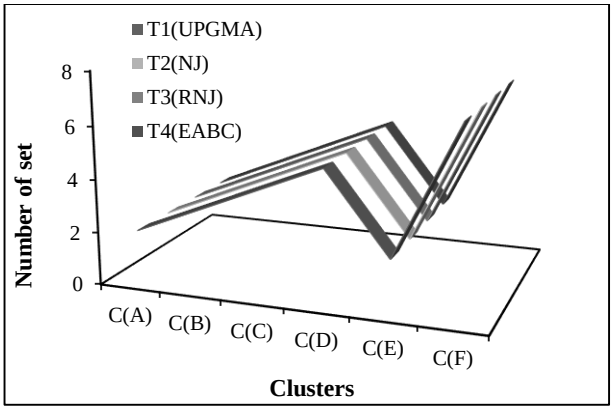


Fig.2. Dissimilarity between *T*1, *T*2, *T*3 and *T*4

6. CONCLUSION

Basically, constructing phylogenetic tree is considered as an NP-complete problem. To solve these kinds of problems, we can use any bio-inspired algorithms. Within these algorithms, swarm intelligence has become a research interest to many scientists in related fields in recent years.

In this paper, the EABC algorithm is proposed suitably for constructing PT, which is an NP-complete problem. This paper proposed an algorithm to promote the convergence rate of the ABC algorithm in such a way it yields an optimal solution and recommended a new formula for searching solution. Along with this, a searching step has been included so that it constructs the optimal solution tree faster.

For experiment two set of the standard dataset have been taken. The results show that even when some algorithm like GA converges faster than other algorithms, the EABC algorithm obtains optimal solution tree the UPGMA, NJ, RNJ and ABC, GA. As computational time depends on the number of sequences and its length, when the length of sequences are decreased, the proposed algorithm takes less time than other mentioned algorithms yet constructing an approximate phylogenetic tree. The dissimilarity of the constructed PT using different algorithms is measured using RF distance, which reveals all the trees are in the same plane.

As a future work, this proposed algorithm can be parallelized to reduce the computational time when large scale sequences are used.

ACKNOWLEDGEMENTS

The authors would like to thank the Ministry of Human Resource Development, Government of India for providing financial support for this research.

REFERENCES

[1] Daniel H. Huson, Vincent Moulton and Mike Steel, "Special Section: Phylogenetics", *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Vol. 6, No. 1, pp. 4-6, 2009

[2] Savarimuthu Ignacimuthu, "*Basic Bioinformatics*", Alpha Science International Ltd., 2005.

[3] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proceedings of IEEE International Conference on Neural Networks*, Vol. 4, pp. 1942-1948, 1995.

[4] Daniel H. Huson, Regula Rupp and Celine Scornavacca, "*Phylogenetic Networks: Concepts, Algorithms and Applications*", Cambridge University Press, 2010.

[5] Dervis Karaboga, "An Idea based on Honey Bee Swarm for Numerical Optimization", Technical report-TR06, Computer Engineering Department, Erciyes University, 2005.

[6] Anabel Martínez-Vargas and Ángel G. Andrade, "Comparing particle swarm optimization variants for a cognitive radio network", *Applied Soft Computing*, Vol. 13, No. 2, pp. 1222-1234, 2013.

[7] Walter M. Fitch and Emanuel Margoliash, "Construction of phylogenetic trees", *Science*, Vol. 155, No. 3760, pp: 279-284, 1967.

[8] Peter H.A. Sneath and Robert R. Sokal, "*Numerical Taxonomy. The Principles and Practice of Numerical Classification*", W. H. Freeman & Co Ltd, 1973.

[9] Naruya Saitou and Masatoshi Nei, "The neighbor-joining method: a new method for reconstructing phylogenetic tree", *Molecular Biology and Evolution*, Vol. 4, No. 4, pp. 406-425, 1987.

[10] Ando, Shin and Hitoshi Iba, "Ant algorithm for construction of evolutionary tree", *Proceedings of the World on Congress on Computational Intelligence*, Vol. 2, pp. 1552-1557, 2002.

[11] M. Kumnorkaew, K. Ku and P. Ruenglertpanyakul, "Application of ant colony optimization to evolutionary tree construction", *Proceedings of 15th Annual Meeting of the Thai Society for Biotechnology*, 2004.

[12] Hui-Ying Lv, Wen-Gang Zhou and Chun-Guang Zhou, "A discrete particle swarm optimization algorithm for phylogenetic tree reconstruction", *Proceedings of International Conference on Machine Learning and Cybernetics*, Vol. 4, pp. 2650-2654, 2004.

[13] Mauricio Perretto and Heitor Silvério Lopes, "Reconstruction of phylogenetic trees using the ant colony optimization paradigm", *Genetics and Molecular Research*, Vol. 4, No. 3, pp. 581-589, 2005.

[14] S June Oh, Je-Gun Joung, Jeong-Ho Chang and Byoung-Tak Zhang, "Construction of phylogenetic trees by kernel-based comparative analysis of metabolic networks", *BMC Bioinformatics*, Vol. 7, No. 1, 2006.

[15] Jason Evans, Luke Sheneman, and James Foster, "Relaxed neighbor joining: a fast distance-based phylogenetic tree construction method", *Journal of Molecular Evolution*, Vol. 62, No. 6, pp. 785-792, 2006.

[16] Ling Qin, Yixin Chen, Yi Pan and Ling Chen, "A novel approach to phylogenetic tree construction using stochastic optimization and clustering", *BMC Bioinformatics*, Vol. 7, 2006.

[17] Priyank Raj Katariya and Sathish S. Vadhiyar, "Phylogenetic predictions on grids", *Proceedings of IEEE 8th International Conference on E-Science*, pp. 58-65, 2009.

[18] Pankaj Bhambri and O.P. Gupta, "Development of phylogenetic tree based on Kimura's Method", *Proceedings of IEEE International Conference on Parallel Distributed and Grid Computing*, pp. 721-723. 2012.

[19] Phylogenetic Tree: Example, Available at: http://science.kennesaw.edu/~jdirnber/Bio2108/Lecture/LecPhylogeny/LecPhylogeny.html

[20] Xin-She Yang, "Engineering optimizations via nature-inspired virtual bee algorithms", *Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach*, pp. 317-323, Vol. 3562, 2005.

[21] Dervis Karaboga and Bahriye Akay, "A comparative study of artificial bee colony algorithm", *Applied Mathematics and Computation*, Vol. 214, No. 1, pp. 108-132, 2009.

[22] R. Srinivasa Rao, S. V. L. Narasimham and M. Ramalingaraju, "Optimization of distribution network configuration for loss reduction using artificial bee colony algorithm", *International Journal of Electrical Power and Energy Systems Engineering*, Vol. 1, No. 2, pp. 116-122, 2008.

[23] Dervis Karaboga and Bahriye Basturk, "An artificial bee colony (ABC) algorithm for numeric function optimization", *IEEE Swarm Intelligence Symposium*, pp. 12-14. 2006.

[24] Karaboga, Dervis, and Bahriye Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm", *Journal of global optimization*, Vol. 39, No. 3, pp: 459-471, 2007.

[25] John H. Holland, "*Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*", U Michigan Press, 1975.

[26] P. Larrañaga, C.M.H. Kuijpers, R.H. Murga, I. Inza and S. Dizdarevic, "Genetic algorithms for the travelling salesman problem: A review of representations and operators", *Artificial Intelligence Review*, Vol. 13, No. 2, pp. 129-170, 1999.

[27] Gusfield, Dan, "*Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*", Cambridge university press, 1997.

[28] p53Dataset: Available at: http://www.bioinformatics.org/p53/nucleotide.html

[29] Dataset: Treezilla, Available at: http://bioinformatics.hungry.com/clearcut/treezilla.fasta

[30] Tetsuo Asano, Jesper Jansson, Kunihiko Sadakane, Ryuhei Uehara and Gabriel Valiente, "Faster computation of the robinson-foulds distance between phylogenetic networks", *Information Sciences*, Vol. 197, pp. 77-90, 2012.