Optimization of Gene Expression Data By Cuckoo Search

S.Dhivya[1], S.V.Evangelin Sonia[2], L.Priyangadevi[3]
[1, 2, 3] Department of Computer Science Engineering
[1, 2] Sri Shakthi Institute of Engineering and Technology
[3] KSR Institute for Engineering and Technology

Abstract*- Microarray gene expression data play a major role in biological processes and systems including evolution, gene regulation and disease mechanism. Biclustering in gene expression data is a subset of the genes demonstrating consistent patterns over a subset of the conditions. The proposed work finds the significant biclusters in large expression data using Cuckoo Search (CS). The cuckoo imitates its egg similar to host bird's egg using levy flight. The proposed work is tested on two benchmark datasets with Cuckoo search with Levy flight (CS) algorithm.*

Keywords*-* Biclustering; Cuckoo search; Levy flight; Gene expression data.

I. INTRODUCTION

DNA microarray technology measures the gene expression level of thousand of genes under multiple experimental conditions (Lockhart and Winzeler, 2000). The conditions may belongs to different time points or different environmental conditions. In some other cases the conditions may have come from different organs, cancerous tissues, healthy tissues, or different individuals. After the number of preprocessing steps, the low level microarray analysis of a microarray can be represented as a numerical matrix. In this matrix the rows represent different genes and columns represent experimental conditions. The row vector of a gene is called the expression pattern of the gene and a column vector is called the expression profile of the condition. Each element of this matrix represents the expression level of a gene under a specific condition, and is represented by a real number. It is usually the logarithm of the relative profusion of the mRNA of the gene under the specific condition. Figure 1 shows the gene expression matrix.

Given a gene expression matrix a common analysis goal is to group genes and conditions into subsets that convey biological significance. In its most common form, this task translates to the computational problem known as clustering.

Formally, for a given set of objects with the vector of attributes for each object, then the clustering aims to partition the object into disjoint classes. So that the objects within a cluster are similar and the objects of disjoint clusters are dissimilar.

For example, when analyzing a gene expression matrix clustering may be applied to the genes for identifying groups of co-regulated genes or cluster the conditions for discovering groups of similar conditions.

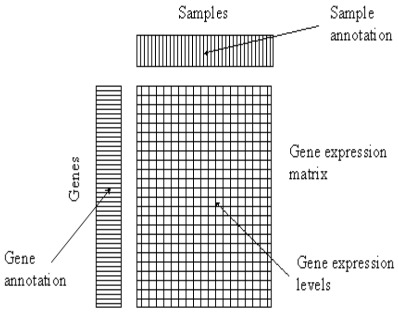


Fig. 1. Gene expression matrix

Analysis via clustering makes several assumptions that may not be completely adequate in all situations. First the clustering can be applied to either genes or conditions; it implicitly directs the analysis to a particular aspect of the system. Second, clustering algorithms usually seek a disjoint cover of the set of elements, requiring that no gene or sample belongs to more than one cluster.

The concept of a bicluster rises to a more flexible computational framework. For example if two genes are related they can have similar expression patterns under certain conditions; similarly, for two related conditions, some genes may exhibit different expression patterns. As a result, each cluster may involve only a subset of genes and a subset of conditions. Biclustering is a simultaneous clustering of both rows and columns of a gene expression data. That is a bicluster is a submatrix spanned by a set of genes and a set of conditions.

The problem of finding a partition of a set of objects into k groups which optimizes a stated condition of partition adequacy is not given as straightforward. Given n objects, the number of ways in which these objects can be partitioned into k non–empty subsets is (Liu. 1968) given in equation (1).

$$P(n,k)=\frac{1}{k!}\sum_{j=0}^{k} {k \atop j} (\ \ 1)^j (k-j)^n \qquad (1)$$

Equation (2) approximates the equation (1)

$$P(n,k)\approx\frac{k^n}{k!}\approx k^{n-k}e^k\sqrt{2\pi k} \qquad (2)$$

Therefore, when the number of clusters k is not known in advance then the total number of valuations is given in equation (3)

$$T(n)=\sum_{k=1}^{n}P(n,k) \qquad (3)$$

Finding significant biclusters in a microarray is a much more complex problem than clustering (Divina and Aguilar-Ruiz, 2006) and it is a NP-hard problem (Tanay et al., 2002). The problem of finding a consistent biclustering can be formulated as an optimization problem. An optimization problem is a problem which determines the set of potential solutions to the problem and defines one or more criteria which measure the quality of individual solution. Solution is obtained by extracting the best solution from the set or an adequately high quality among the set. For a finite unimodal optimization problem, the basic algorithmic solution typically assesses exhaustively as many solutions as needed in the search space to prove a given solution is at least better than any other solution in the search space. This is the optimum solution returned by the algorithm. Let S be a set of solutions to a problem, and let $f : S \rightarrow R$ be an objective function to be minimized and that measures the quality of these solutions then the optimal solution $m \in S \mid \forall s \in S;\ f(m) < f(s)$.

This work develops and implements the biclustering based on the most popular and robust bio inspired strategy Cuckoo Search (CS). In the CS, each nest consists of single egg and cuckoo imitates egg using Levy flight. The remainder of this paper is organized as follows: Section 2 provides the structure of bicluster and related works in biclustering. Section 3 gives a general overview of the Cuckoo Search. Section 4 presents the detailed experimental setup and results.

II. REVIEW OF RELATED WORKS

Cheng & Church (2000) presented first biclustering approach for gene expression data. Their algorithm adopts a sequential covering strategy in order to return a list of n biclusters from an expression data matrix. Tanay et al. (2002) introduced Statistical-Algorithmic Method for Bicluster Analysis (SAMBA), a biclustering algorithm that performs simultaneous bicluster identification by using exhaustive enumeration.

Murali and Kasif (2003) aimed at finding conserved gene expression motifs (xMOTIFs). They defined an xMOTIF as a subset of genes that is simultaneously conserved across a subset of the conditions. Ben-Dor et al. (2003) defined a bicluster as an Order-Preserving Sub-Matrix (OPSM). Bergmann et al. (2003) proposed Iterative Signature Algorithm (ISA) and provides a definition of biclusters as transcription modules to be retrieved from the expression data.

Mitra and Banka (2006) presented a Multi-Objective Evolutionary Algorithm (MOEA) based on Pareto dominancy. Divina & Aguilar-Ruiz (2006) presented a Sequential Evolutionary BIclustering (SEBI) approach.The term sequential refers the way in which bicluster are discovered, only one bicluster obtained per each run of the evolutionary algorithm.

Liu & Wang (2007) introduced Maximum Similarity Bicluster (MSB) algorithm. DiMaggio et al. (2008) proposed an approach that is based on the optimal re-ordering of the rows and columns of a data matrix so as to globally minimize dissimilarity metric. Liu et al. (2009) based their biclustering approach on the use of a PSO together with crowding distance as the nearest neighbor search strategy, which speeds up the convergence to the Pareto front and also guarantee diversity of solutions. Coelho et al. (2009) presented an immune-inspired algorithm for biclustering based on the concepts of clonal selection and immune network theories adopted in the original aiNet algorithm.

Ayadi et al. (2012) proposed as a Pattern-Driven Neighborhood Search (PDNS) approach for the biclustering problem. Huang et al. (2012) proposed a new biclustering algorithm based on the use of an Evolutionary Approach (EA) together with hierarchical clustering. Ray et al. (2013) introduced a CoBi: Pattern Based Co-Regulated Biclustering of gene expression Data. It is mainly used for grouping both positively and negatively regulated genes from microarray expression data.

III. CUCKOO SEARCH (CS) WITH LEVY FLIGHT

Cuckoo search is an optimization technique developed by Xin-She Yang and Suash Deb in (2009) based on the brood parasitism of cuckoo species by laying their eggs in the nests of other host birds. Based on the selfish gene theory (Dawkins, 1989) this parasitic behavior increases the chance of survival of the cuckoo's genes since the cuckoo needs not spend any energy rearing its young one. It allows the cuckoo to spend more time for breeding and laying more eggs. If a host bird discovers the eggs which are not their own, it will either throw these foreign eggs away or simply abandon its nest and build a new nest elsewhere. The CS algorithm utilizes these behaviors in order to traverse the search space and find optimal solutions.

A set of nests with one egg inside are placed in random locations in the search space where the eggs each represent a candidate solution. Numbers of cuckoos are assigned to traverse the search space recording the highest objective values for different encountered candidate solutions. The cuckoos utilize a search pattern called Levy flight which is encountered in real insects, fish, birds and grazing animals (Viswanathan et al. 2005).

The rules for CS are described as follows:

- Each cuckoo lays one egg at a time, and dumps it in a randomly chosen nest
- The best nests with high quality of eggs will carry over to the next generations;
- The number of available host nests is fixed, and a host can discover a foreign egg with a probability $p_a \in [0, 1]$. In this case, the host bird can either throw the egg away or abandon the nest so as to build a completely new nest in a new location.

Algorithm 1 : Pseudo code for Cuckoo Search with Levy flight

```
Generate an initial population of n host nests;
while (t<MaxGeneration) or (stop criterion)

Get a cuckoo randomly (say, i) and replace its solution by performing Levy flights;
    Evaluate its fitness $F_i$
    Choose a nest among n (say, j) randomly;
    if ($F_i < F_j$)
        Replace j by the new solution;
    end if
A fraction ($p_a$) of the worse nests is abandoned and new ones are built;
    Keep the best solutions/nests;
    Rank the solutions/nests and find the current best;
    Pass the current best to the next generation;
  end while
```

The traditional CS algorithm uses fixed value for both p_a and α . These values are set in the initialization step and cannot be changed during new generations. The main drawback of this method appears in the number of iterations to find an optimal solution. If the value of p_a is small and the value of α is large, the performance of the algorithm will be poor and leads to considerable increase in number of iterations. If the value of p_a is large and the value of α . is small, the speed of convergence is high but it may be unable to find the best solutions.

The key difference between the ICS and CS is in the way of adjusting p_a and α . To improve the performance of the CS algorithm and eliminate the drawbacks lies with fixed values of p_a and α , the ICS algorithm uses variables p_a and α .In the early generations, the values of p_a and α must be big enough to enforce the algorithm to increase the diversity of solution vectors. However, these values should be decreased in final generations to result in a better fine-tuning of solution vectors. The values of p_a and α . are dynamically changed with the number of generation.

The improved cuckoo search algorithm uses a balanced combination of a local random walk and the global explorative random walk, controlled by a switching parameter pa. The local random walk can be written as:

$$x_i^{t+1} = x_i^t + \alpha s \otimes H(p_a - \epsilon) \otimes (x_j^t - x_k^t)$$

where x^t_j and x^t_k are two different solutions selected randomly by random permutation, H(u) is a Heaviside function,is a random number drawn from a uniform distribution and s is the step size. On the other hand, the global random walk is carried out by using Le´vy flights

$$x_i^{t+1} = x_i^t + \alpha L(s, \lambda)$$

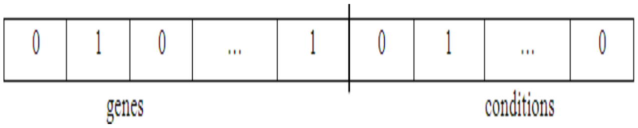
Where

$$L(s,\lambda) = \frac{\lambda \Gamma(\lambda) \sin(\pi\lambda/2)}{\pi} \frac{1}{s^{1+\lambda}}, (s >> s_0 > 0)$$

Here, $\alpha > 0$ is the step size scaling factor, which should be related to the scales of the problem of interest. In most cases, we can use α = O(L/10), where L is the characteristic scale of the problem of interest, while in some cases α = O(L/100) can be more effective and avoid flying too far. The above equation is essentially the stochastic equation for a random walk. In general, a random walk is a Markov chain whose next status/location only depends on the current location (the first term in the above equation) and the transition probability (the second term). However, a substantial fraction of the new solutions should be generated by far field randomization and their locations should be far enough from the current best solution; this will make sure that the system will not be trapped in a local optimum

A. Biclustering representation

Each bicluster is encoded as an egg in the nest. Each egg is fixed length of size m+n, where m and n are the number of genes and conditions of the microarray dataset respectively. The first m bits represent m genes and the following n bits represent n conditions. Each bicluster is represented by a fixed sized binary string called an egg, with a bit string for genes attached with another bit string for conditions. An egg represents a candidate solution for this optimal bicluster generation problem. A bit is set to one if the corresponding gene and/or condition is present in the bicluster, and reset to zero otherwise. Figure 3 shows an encoded representation of a bicluster.



0	1	0	...	1	0	1	...	0

Fig. 2. Encoding representation of a Bicluster

The Cuckoo search works well for continuous optimization problem. So the individual dimension of an egg is represented by a real number. The mapping function for an egg into a binary string representation of a bicluster is given in equation (6) as follows:

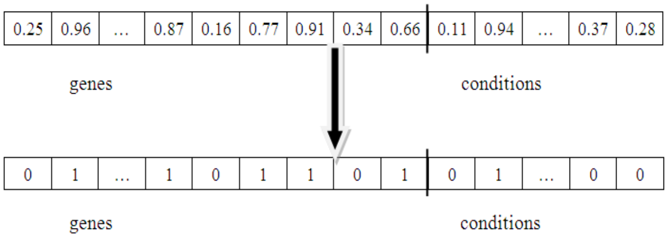
$$y_{ij} = \begin{cases} x_{ij} < 0.5 & 0 \\ \text{otherwise} & 1 \end{cases} \qquad (6)$$

where

x_{ij} - Random value generated for jth gene/condition of ith egg.

y_{ij} - Binary string representation of bicluster of x_{ij}

In y_{ij}, if a bit is set to 1 then the corresponding gene or condition belongs to the encoded bicluster; otherwise it is not. Figure 3 shows the representation of an egg and its mapped bicluster representation.



0.25	0.96	...	0.87	0.16	0.77	0.91	0.34	0.66	0.11	0.94	...	0.37	0.28
0	1	...	1	0	1	1	0	1	0	1	...	0	0

Fig. 3. Representation of an egg and its mapping to Bicluster

B. Fitness function

Mean Squared Residue (MSR) problem has been proposed by Cheng and Church (2000) for identifying biclusters. Let gene expression data matrix A has M rows and N columns where a cell a_{ij} is a real value that represents the expression level of gene i under condition j. Matrix A is defined by its set of rows, R = {r1, r2, ..., rM} and its set of columns C = {c1, c2, ..., cN}. Given a matrix, biclustering finds sub-matrices that are subgroups of genes and subgroups of conditions, where the genes exhibit highly correlated behavior for every condition. Given a data matrix A, the goal is to find a set of biclusters such that each bicluster exhibits some similar characteristics. Let A_{IJ} = (I, J) represent a submatrix of A where $I \in R$ and $J \in C$. A_{IJ} contains only the elements a_{ij} belonging to the submatrix with set of rows I and set of columns J. The concept of bicluster was introduced by Cheng and Church (2000) to find correlated subsets of genes and a subset of conditions.

Let a_{iJ} denote the mean of the i-th row of the bicluster (I, J), a_{Ij} the mean of the j-th column of (I, J), and a_{IJ} the mean of all the elements in the bicluster. As given in more formally,

$$a(I,j)=\frac{1}{|J|}\sum_{j\in J} a_{i,j}$$

$$a(i,J)=\frac{1}{|I|}\sum_{i\in I} a_{i,j}$$

$$a(I,J)=\frac{1}{|I||J|}\sum_{i\in I}\sum_{j\in J} a_{i,j}$$

The residue of an element a_{ij} in a submatrix A_{IJ} equals

$$r_{i,j} = a_{i,j} + a_{I,J} - a_{I,j} - a_{i,J}$$

The difference between the actual value of a_{ij} and its expected value predicted from its row, column and bicluster mean are given by the residue of an element. It also reveals its degree of coherence with the other entries of the bicluster it belongs to. The quality of a bicluster can be evaluated by computing the MSR H, i.e. the sum of all the squared residues of its elements is given in equation (7).

$$H(I,J)=\frac{1}{|I||J|}\sum_{i\in I}\sum_{j\in J} r_{i,j}^{2} \qquad (7)$$

The lowest score of H(I,J) is 0 which indicates the gene expression levels vary in harmony. This includes the trivial or constant biclusters where there is no fluctuation. These trivial biclusters may not be interesting but need to be revealed and masked so more interesting ones can be found. The gene variance may be a complementary score to reject trivial biclusters. The row gene can be represented in equation (8) as follows:

$$\mathrm{Var}_r(I,J) = \frac{1}{|I|}\sum_{i\in I} v_r(i) \qquad (8)$$

$$v_r(i) = \frac{1}{|J|}\sum_{j\in J}\left(a_{i,j} - a_{i,J}\right)^2$$

The optimization task is finding one or more biclusters by maintaining the two competing constraints, viz., homogeneity and gene variance. The fitness function for obtaining bicluster is defined in equation (9) as follows

$$f(I,\ J) = H(I,\ J) + \frac{1}{\mathrm{Var}(I,\ J)} \qquad (9)$$

to be minimized.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

The biclustering algorithm has been applied to data set in order to study its performance , namely the yeast saccharmyces cerevisiae stress expression data (Gasch et al., 2000). The Gasch yeast is the sacchromyces cerevisiae with 2993 genes and 173 conditions. Table 1 shows the parameter and ite value used in this paper.

Table 1: Parameter and its Value

Parameter	Value
p_a	0.3
A	1
Λ	1.5
Number of nests	20
Iteration	100

According to the problem formulation the size of the extracted bicluster should be as large as possible. The bicluster should satisfy two requirements simultaneously. The expression levels of each gene within the bicluster should have low MSR value. The bicluster gene variance should be high. The MSR represents the variance of the selected genes and conditions with respect to homogeneity of the bicluster and gene variance removes the simple bicluster. To quantify biclusters homogeneity and size that satisfy the Coherence Index (CI) is used as the measure of evaluating their goodness. CI is defined as the ratio of MSR score to the size of the formed bicluster. Table 2 shows the sample experimental results obtained for saccharomyces cerevisiae expression data and 5 biclusters are chosen from 20 biclusters. Figure 4 shows the small bicluster of size 10x5 for saccharomyces cerevisiae expression data.

Table 2: Experimental Results for Saccharomyces Cerevisiae Expression Data

Bicluster No.	Genes	Conditions	Volume	MSR	Gene Variance	CI	Fitness
BC_1	1533	94	1441	0.6727	0.6887	2.1246	2.1247
BC_3	1505	91	1369	0.6445	0.6583	2.1634	2.1634
BC_5	1477	87	1284	0.6448	0.6671	2.1437	2.1437
BC_8	1482	77	1141	0.6390	0.6631	2.1469	2.147
BC_{10}	1486	92	1367	0.6611	0.6773	2.1375	2.1376

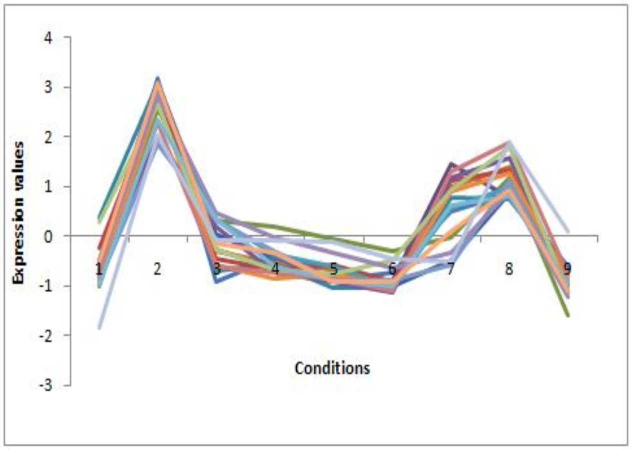


Fig 4. Small bicluster of size 10x5 for saccharomyces cerevisiae expression data

V. CONCLUSION

In this work cuckoo search with levy flight algorithm for biclustering microarray gene expression data is proposed. It focuses maximum biclusters with lower Mean Squared Residue and higher gene variance. Cuckoo Search strategy is applied to find the optimal bicluster using Levy flight. A qualitative assessment of results are provided on yeast saccharomyces cerevisiae stress expression data to demonstrate the effectiveness of the proposed method.

REFERENCES

[1] Yang, XS & Deb, S 2014, ‘Cuckoo search recent advances and applications’, Journal of Neural Computing and Applications, vol. 24, no. 1, pp. 169-174.

[2] Angiulli, F, Cesario, E & Pizzuti, C 2008, ‘Random walk biclustering for microarray data’, Journal of Information Sciences, vol. 178, no. 6, pp. 1479-1497.

[3] Ayadi, W, Elloumi, M & Hao, JK 2012, ‘BiMine+: An efficient algorithm for discovering relevant biclusters of dna microarray data’, Knowledge Based System, vol. 35, no. 11, pp. 224-235.

[4] Bleuler, S, Prelic, A & Zitzler, E 2004, ‘An ea framework for biclustering of gene expression data’, Proceedings of congress on evolutionary computation, vol. 1, pp. 166-173.

[5] Cheng, Y & Church, GM 2000, ‘Biclustering of expression data’, Proceedings of the 8th international conference on intelligent systems for molecular biology, pp. 93-103.

[6] Cho, RJ, Campbell, MJ, Winzeler, EA, Steinmetz, L, Conway, ALW, Wolfsberg, T, Gabrielian, A, Landsman, D, Lockhart, D & Davis, R 1998, ‘Agenome-wide transcriptional analysis of the mitotic cell cycle’ Journal of Molecular Cell, vol. 2, no. 1, pp. 65-73.

[7] Christinat, Y, Wachmann, B & Zhang, L 2008, ‘Gene expression data analysis using a Novel approach to biclustering combining discrete and continuous data’, IEEE/ ACM Transactions on Computational Biology and Bioinformatics, vol.5, no.4, pp. 583-593.

[8] Clough, SJ & Bent, A, F 1998, ‘Floral dip: a simplified method for Agrobacterium-mediated transformation of Arabidopsis thaliana’, The Plant Journal, vol. 16, no. 6, pp. 735-743.

[9] Das, S & Idicula, S 2010, ‘Application of reactive grasp to the biclustering of gene expression data’, Proceedings of the international symposium on bio computing, vol. 18, Suppl. 1, pp. 1-8.

[10] Dharan, A & Nair, A 2009, ‘Biclustering of gene expression data using reactive greedy randomized adaptive search procedure’, BMC Bioinformatics, vol. 10, Suppl. 1, S27.

[11] Duval, B & Hao, JK 2010, ‘Advances in metaheuristics for gene selection and classification of microarray data’, Briefings in Bioinformatics, vol. 11, no. 1, pp. 127–141.

[12] Han, L & Yan, H 2012, ‘Hybrid method for the analysis of time series gene expression data’, Knowledge-Based Systems, Vol. 35, pp. 14-20.

[13] Hauke, J, & Kossowski, T 2011, ‘Comparison of values of pearson’s and spearman’s correlation coefficient on the same sets of data’, Journal of Quaestiones Geographicae, vol. 30, no. 2, pp. 87-93.

[14] Gajavelli, S, Wood, MP, Pennica, D, Whittemore, SR & Tsoulfas, P 2004, ‘BMP signaling initiates a neural crest differentiation program in embryonic rat CNS stem cells’, Journal of Experimental Neurology, vol. 188, no. 2, pp. 205-223.

[15] Gandomi, AH, Yang, XS & Alavi, AH 2013, ‘Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems’, Journal of Engineering with Computers, vol. 29, no. 1, pp. 17-35.

[16] Luan, Y & Li, H 2003, ‘Clustering of time-course gene expression data using a mixed effects model with B-splines”, Journal of Bioinformatics, vol. 19, no. 4, pp. 474-482.

[17] vazoie, S, Hughes, JD, Campbell, MJ, Cho, RJ & Church, GM 1999, ‘Systematic determination of genetic network architecture’, Journal of Nature Genetics , vol. 22 , no. 3, pp. 281-285.

[18] Teng, L & Chan, L 2008, ‘ Discovering biclusters by iteratively sorting with weighted correlation coefficient in gene expression data’, Journal of Signal Processing Systems, vol. 50, no. 3, pp. 267-280.

[19] Valian, E , Mohanna, S & Tavakoli, S 2011, ‘Improved cuckoo search for global optimization’, International Journal of Communications & Information Technology’, vol. 1, no.1, pp. 31-44.

[20] Wen, X, Fuhrman, S, Michaels, GS, Carr, DB, Smith, S, Barker, JL & Somogyi, R 1998, ‘Large-scale temporal gene expression mapping of central nervous system development’, Proceedings of the national academy of sciences of the United States of America, vol. 95, no.1, pp. 334-339.

[21] Yang, XS & Deb, S 2009, ‘Cuckoo search via Levy flights’, Proceedings of world congress on nature & biologically inspired computing, pp. 210-214.

[22] Sureja, N 2012, ‘New inspirations in nature: A Survey’, International Journal of Computer Applications & Information Technology, vol.1, no.3, pp. 21-24.