International Journal of Engineering, Mathematical and Physical Sciences
ISSN: 2517-9934
Vol:8, No:4, 2014

Mining Correlated Bicluster from Web Usage Data Using Discrete Firefly Algorithm Based Biclustering Approach

K. Thangavel, R. Rathipriya

***Abstract*—**For the past one decade, biclustering has become popular data mining technique not only in the field of biological data analysis but also in other applications like text mining, market data analysis with high-dimensional two-way datasets. Biclustering clusters both rows and columns of a dataset simultaneously, as opposed to traditional clustering which clusters either rows or columns of a dataset. It retrieves subgroups of objects that are similar in one subgroup of variables and different in the remaining variables. Firefly Algorithm (FA) is a recently-proposed metaheuristic inspired by the collective behavior of fireflies. This paper provides a preliminary assessment of discrete version of FA (DFA) while coping with the task of mining coherent and large volume bicluster from web usage dataset. The experiments were conducted on two web usage datasets from public dataset repository whereby the performance of FA was compared with that exhibited by other population-based metaheuristic called binary Particle Swarm Optimization (PSO). The results achieved demonstrate the usefulness of DFA while tackling the biclustering problem.

***Keywords*—**Biclustering, Binary Particle Swarm Optimization, Discrete Firefly Algorithm, Firefly Algorithm, Usage profile Web usage mining.

I. Introduction

IN the literature, clustering is the most commonly used data mining technique to extract the hidden usage patterns from the web data. Standard clustering methods such as k-means, hierarchical clustering, self organizing maps [12], [15], [16], [20]. Conventional clustering techniques are based on similarity between users across all pages of a website. However, users may be co- regulated under some specific pages and shows weak similarity beyond these pages. Therefore, a group of user forms cluster under a subset of pages. This technique of two-way clustering referred to as biclustering, in which both users and pages are grouped simultaneously.

Several biclustering algorithms have been proposed in literature for different application. To extract biclusters, these algorithms usually employ heuristic or probabilistic model. An illustrative discussion on many of these algorithms can be found in [10], [17].

Biclustering was reintroduced by Cheng and Church [1] in the domain of gene expression data analysis and it identifies biclusters with the help of mean squared residue score, which is a measure of the coherence of rows and columns in the bicluster. Biclustering techniques have been applied in different contexts, such as in bioinformatics, time series expression data, text mining, and collaborative filtering [4]-[6].

Many different biclustering algorithms can be found in the literature [16]. In particular, due to the highly combinatorial nature of this problem, bio-inspired metaheuristics have been successfully adopted to tackle it, such as genetic algorithms (GA) [8], particle swarm optimization (PSO), artificial immune systems (AIS) [6], and ant colony optimization (ACO) [7]. In this study, to assess the performance of discrete Firefly algorithm in mining coherent and high volume biclusters, a simple modification of the algorithm was performed in order to allow for the representation of binary solutions.

Two datasets namely MSNBC dataset and MSWEB dataset from UCI repository[1], were considered, and the assessment is done here having as standard the levels of performance exhibited by Binary PSO (BPSO). Overall, the results achieved so far suggest that the DFA algorithm is competitive in terms of locating coherent biclusters and shows better computational efficiency.

The rest of the paper is organized as follows: Section II provides a brief account on the biclustering problem. Section III describes the discrete firefly algorithm while Section IV reviews the main steps of the BPSO algorithms. In Section V, the empirical results achieved by DFA and BPSO are discussed. Section VI, provides final remarks.

II. Biclustering: A Preliminaries

In general, the goal of a biclustering algorithm is to locate sizeable and coherent biclusters efficiently hidden in a given data matrix. Madeira and Oliveira [9] devised four dimensions to classify these algorithms: the type of biclusters found; the structure of the biclusters; the method used to identify biclusters; and the context in which they are applied.

Recently, there have been an increasing number of studies investigating the application of bio-inspired algorithms to biclustering [2], [5]-[8], [11], [19]. One crucial issue in this context is to use a good objective function (also called fitness function) to measure the bicluster quality. As in [13], [14], the following fitness function is used in this study.

$$\text{Max } f(u, p) = |u| * |p|$$

K. Thangavel and R. Rathipriya are with the Periyar University, Salem, India (e-mail: drktvelu@yahoo.com, rathi_priyar@yahoo.co.in).

[1] http://archive.ics.uci.edu/ml/datasets.html

Subjected to $g(u,p) \geq \delta$ (1)

where $g(u,p) = max\left\{\frac{\sum_{i=1}^{n}\sum_{j=1}^{n}|r\_row_{ij}|-n}{n^2-n}, \frac{\sum_{k=1}^{m}\sum_{l=1}^{m}|r\_col_{kl}|-m}{m^2-m}\right\}$

$r\_row_{?j}$ is the correlation between row i and row j, r_col_{kl} and is the correlation between column k and column l. and $\boldsymbol{\delta}$ is a threshold to be pre-defined by the end user.

Equation (1) is adopted here to measure the quality of the biclusters produced by DFA and BPSO algorithms. This function makes use of the Average Correlation value of a bicluster (ACV), proposed by Teng and Chan [18] to measure the bicluster coherence.

III. Firefly Optimization Algorithm

Firefly Optimization algorithm (FA), developed by Xin-She Yang [21], is inspired by the light attenuation over the distance and fireflies' mutual attraction, rather than by the phenomenon of the fireflies' light flashing. Algorithm considers what each firefly observes at the point of its position, when trying to move to a greater light-source, than his own. The firefly meta-heuristic relies on a set of artificial fireflies which communicate with each other to solve optimization problems. The behavior of artificial fireflies is modeled according to the behavior of fireflies in nature, which search for a mating partner by emitting a flashing light.

For simplicity, summarize these flashing characteristics as the following three rules:

a. All fireflies are unisex, so that one firefly is attracted to other fireflies regardless of their sex.
b. Attractiveness is proportional to their brightness, thus for any two flashing fireflies, the less bright one will move towards the brighter one. The attractiveness is proportional to the brightness and they both decrease as their distance increases. If no one is brighter than a particular firefly, it moves randomly.
c. The brightness of a firefly is affected or determined by the landscape of the objective function to be optimized. For a maximization problem, the brightness can simply be proportional to the value of the objective function.

Based on these three rules, the basic steps of the firefly algorithm (FA) can be summarized as the pseudo code shown in Algorithm 1.

A. Discrete Firefly Algorithm Based Biclustering Approach

This section presents mapping the concepts of the firefly meta-heuristic to the problem of biclustering. Just as the real fireflies search for a mating partner by means of flashing lights, have a number of artificial fireflies which search for the optimal bicluster solution. Thus, map the attraction behavior of fireflies to the problem of selecting the optimal bicluster as follows:

1. a firefly becomes an artificial firefly
2. the position of a firefly becomes a bicluster solution
3. the brightness of a firefly becomes the quality of a bicluster solution evaluated with a fitness function
4. the attractiveness between two fireflies becomes the hamming distance between two biclusters
5. the movement of a firefly is mapped to a modification of the firefly's current position (i.e. bicluster)

Algorithm 1: Firefly Algorithm

1. Initialize the initial population of fireflies X_i, i = 1, 2 ..., m.
2. Define light absorption coefficient λ (control parameter)
3. while (t < MaxGeneration)
for i = 1 to m-1
for j = i +1 to m
if fitness(X_i) > fitness(X_j)
Move X_i toward X_j;
End if
Attractiveness varies with distance r via $e^{-\lambda r}$
Evaluate fitness of new solutions
End for j
End for i
Rank the fireflies
End while
4. Return the best fireflies

Initial Population

Each individual in the chosen algorithms (BPSO and DFA) denotes a candidate bicluster and is formed by two strings of sizes *n* and *m*, where *n* and *m* denote the total number of rows and columns of the data matrix [10]. The binary encoding is adopted in which a bit is set to one(1) when the corresponding user or page is included in the bicluster otherwise it is set to zero (0). In this study, initial population is generated randomly using random function in Matlab toolbox.

1	1	0	0	…	1	0	0	0	1	1	0	0	…	1	1
			Users									Pages			

Fig 1 A binary encoded bicluster

1) Directed Movement

In standard Firefly Optimization algorithm, firefly movement is based on light intensity and comparing it between each two fireflies. Thus for any two fireflies, the less bright one will move towards the brighter one. In Discrete Firefly algorithm, the movement is caused by the use of genetic crossover operator, and that is given as follows:

$$r = \text{Hamming similarity}(x_i, x_j)$$
$$x_i = \text{crossover}(x_i, x_j, r)$$

where Hamming similarity(.) gives similarity between two fireflies x_i, x_j and r is used as crossover probability rate. But firefly x_i new position causes better cost, it will move to that new position. New firefly algorithm can be summarized as the pseudocode is shown in Algorithm 2. This strategy makes a social behavior for all fireflies and they move towards global best.

2) Crossover Operator

The basic operator for producing new solutions in the GA is that of crossover. Like its counterpart in nature, crossover produces new individuals that have some parts of both parent's genetic material. In this study, Multi-point Crossover operator is used to move firefly towards brighter fly.

For multi-point crossover, mp crossover positions mp = {l_1, l_2, ..l_{mp}} are selected randomly and n + m is the length of the binary string. The bits between successive crossover points are exchanged between the two strings to produce two new offspring.

The idea behind multi-point, and indeed many of the variations on the crossover operator, is that the parts of the binary string representation that contribute to the most to the performance of a particular individual may not necessarily be contained in adjacent substrings [4]. Further, the disruptive nature of multi-point crossover appears to encourage the exploration of the search space, rather than favoring the convergence to highly fit individuals (biclusters) early in the search, thus making the search more robust.

Algorithm 2: Discrete Firefly Optimization based Biclustering Algorithm

1. *Initialize the binary encoded biclusters as initial fireflies* $X= \{x_1, x_2, \ldots, x_n\}$
2. *Do*

    for i = 1: N

    for j =1 : i

    if fitness (x_i*) < fitness (*x_j*) then*

    r = Compute_Distance (x_i, x_j*)*

    x prob= r / number of bits in x_i

    x_i*= Crossover(*x_i, x_j*, xprob)*

    end if

    end for

    end for

    until (Stopping_Condition())
3. *Return the best bicluster as the optimal usage profile*

In the first step of the proposed algorithm, each firefly is associated with a generated bicluster solution. These initial solutions are further improved in an iterative process which stops when the best solution has been the same over the last iterations (*Stopping_Condition*).

In each iteration, if the fitness of the solution associated to a firefly is better than the fitness of the solution associated to another firefly it means that the latter firefly will be attracted towards the first one and thus it will have its solution improved.

IV. Binary PSO: A Contestant Algorithm

In the sequel, main steps of Binary PSO (BPSO) algorithm are outlined briefly as investigated here for biclustering purposes. Particle Swarm Optimization (PSO) [9] method for optimization was first introduced by Kennedy and Eberhart and is inspired by the emergent motion of a flock of birds searching for food. It is a population-based optimization tool, which could be implemented and applied easily to solve various function optimization problems. It is used to explore the search space of a given problem to find the settings or parameters required to maximize a particular objective.

Each particle knows its best value so far (pbest) and its position. This information is analogy of personal experiences of each particle. Moreover, each particle knows the best value so far in the group (gbest) among pbests. This information is analogy of knowledge of how the other particles around them have performed. Each particle tries to modify its position based on current positions, current velocities, distance between the current position and pbest and distance between the current position and gbest.

The PSO algorithm consists of three steps, which are repeated until stopping criteria is met :

Step 1. Evaluate the fitness of each particle

Step 2. Update personal best (pbest) of each particle, and global best (gbest)

Step 3. Update velocity and position of each particle

Fitness evaluation is conducted by supplying the candidate solution to the objective function or fitness function. Personal best (pbest) of each particle, global best(gbest) and positions are updated by comparing the newly evaluated fitness against the previous individual's pbest and gbest.

The velocity and position update step is responsible for the optimization ability of the PSO algorithm. The velocity of each particle is updated using the following equation:

$$vi(t + 1) = w * vi(t) + c1r1[pbest(x(t)) − xi(t)] + c2r2[gbest(t) − xi(t)] \quad (2)$$

$$xi(t+1) = xi(t) + vi(t + 1) \quad (3)$$

The index of the particle is represented by i. Thus, vi(t) is the velocity of particle i at time t and xi(t) is the position of particle i at time t. The parameters w, c1, and c2 are user supplied coefficients. The values r1 and r2 are random values regenerated for each velocity update. The value pbest(x(t)) is the individual best candidate solution for particle i at time t, and g(t) is the particle's global best candidate solution at time t. w is inertia weight and c1 and c2 determine the relative influence of the social and cognitive components. Equation 3 is used to update the position of the particle. This process is repeated until the best solution is found or terminate conditions are satisfied.

However, the typical PSO is designed for continuous function optimization problems; it is not designed for discrete function optimization problems. Modified version of PSO called Binary Particle Swarm Optimization (BPSO) [15], [16], [17] that can be used to solve discrete function optimization problems.

The major difference between binary PSO and typical PSO is that the velocities and positions of the particles are defined in terms of the changes of probabilities and the particles are formed by integers in {0, 1}. Therefore, a particle flies in a search space restricted to zero and one. The speed of the particle must be constrained to the interval [0, 1]. A logistic sigmoid transformation function S(vi(t + 1)) is shown in (4) can be used to limit the speed of particle.

$$S(Vi(t+1)) = \frac{1}{1+e^{\hat{}}(Vi(t+1)} \quad (4)$$

The new position of the particle is obtained using (5) shown below:

$$x_i = \begin{cases} 1, \text{ if } \; r_3 < S(v_i(t+1)) \\ 0 \quad \text{otherwise} \end{cases} \quad (5)$$

where r_3 is a uniform random number in the range [0, 1].

V. Experimental Analysis

To assess the applicability of DFA in mining coherent and larger size biclusters, experiments were conducted on two UCI repository datasets namely MSNBC and MSWEB* whose general purpose is to perform automated suggestions for a given user based on the opinions of other users with similar interests. The value for the coherence threshold δ adopted for the two datasets was 0.95.

A. Dataset Description

The MSWEB dataset was obtained from the UCI KDD archive http://kdd.ics.uci.edu/databases/msweb/msweb.html) and records the areas within www.microsoft.com that users visited in one-week time frame during February 1998. Two separate datasets are provided, a training set and a test set. After data preprocessing, the filtered data consisted of 32711 sessions and 285 pages. Further preprocessed MSWEB dataset where the root pages were considered in the pageview of a session. This preprocessing step resulted in total of 20 categories namely “library”, “developer”, “home”, “finance”, “repository”, “gallery”, “catalog”, “mail”, “ads”, “education”, “magazine”, “support”, “ms”, “technology”, “search”, “country”, “business”, “entertainment”, “news”, “feedback”. These page view categories are numbered sequentially from 1 to 20 respectively.

The second dataset (MSNBC) from the UCI dataset repository (http://kdd.ics.uci.edu/) that consists of Internet Information Server (IIS) logs for msnbc.com and news-related portions of msn.com for the entire day of September 28, 1999 (Pacific Standard Time). Each sequence in the dataset corresponds to page views of a user during that twenty-four hour period. Each event in the sequence corresponds to a user’s request for a page. Requests are not recorded at the finest level of detail but they are recorded at the level of page categories as determined by the site administrator. There are 17 page categories, namely, “front page,” “news,” “tech,” “local,” “opinion,” “on-air,” “misc,” “weather,” “health,” “living,” “business,” “sports,” “summary,” “bbs” (bulletin board service), “travel,”“msn-news,” and “msn-sports” and they are numbered from 1 to 17 sequentially.

For the purpose of this paper, assume that each user session is viewed as a single transaction. The session file may be filtered to remove very small transactions and very low support references (i.e., URL references that are not supported by a specified number of user transactions). This type of support filtering can be important in removing noise from the data. Finally, MSWEB dataset contains 5024 user sessions and MSNBC dataset contains 3386 user sessions (i.e. transaction)

The algorithms were finally configured as follows for both datasets:

a) The population size was 100 individuals (either fireflies or particles)
b) The termination condition was to run through 2,000 iterations
c) In DFA, crossover rate was set as 0.5
d) The PSO algorithm was configured with the social and cognitive acceleration coefficients c1 and c2 set both as 2.0 and variable inertia weight (W) is used to set the balance between the global and local search abilities.

$$W = wmax - ((wmax - wmin) / itermax) * iter \quad (6)$$

where wmin = 0.4, wmax =0.9, itermax is the maximum number of iteration and iter is the current iteration in progress

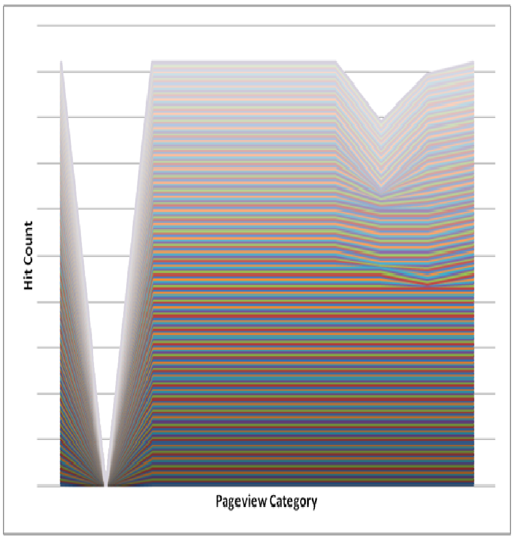
Degree of overlapping, percentage of users covered in the biclusters called user coverage , percentage of pages covered in the biclusters called page coverage, mean volume are the quantitative indices defined by Das C. et al. (2008) [3] to evaluate quantitatively the quality of generated biclusters. Mean ACV is calculated using the following

$$\text{Mean ACV}=\frac{\sum_{bic=1}^{nb} ACV\ (B_{bic})}{nb} \quad (7)$$

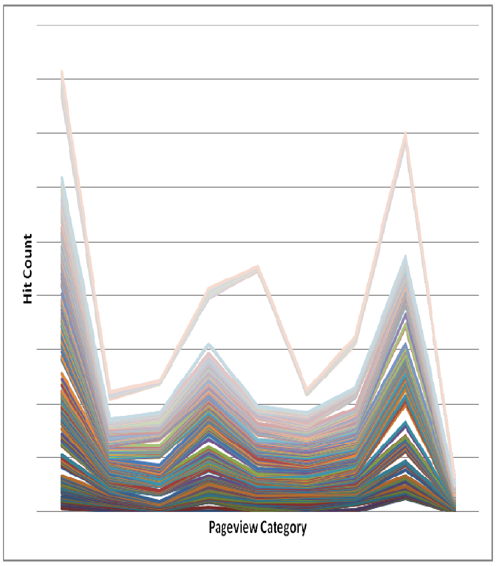
where nb is the total number of biclusters (here nb=100) and Bbic is the bicth bicluster

Tables I and II provide a comparison of the average performance achieved in 2,000 runs by the two algorithms, considering the same metrics as defined before. The results confirm that DFA was certainly a good choice for mining coherent biclusters (i.e., with ACV values greater than the predefined δ) in both datasets, significantly outperforming BPSO with regard to average ACV and volume for the MSNBC and MSWEB datasets. Overall, DFA has prevailed over the others in terms of the criteria of locating the best solution and rate of overlapping, meaning that this algorithm is computationally faster, has a good convergence rate.

The volume, ACV, number of users and pages of optimal bicluster extracted from each dataset using DFA and BPSO based biclustering algorithm are tabulated in Table III. It is clear from the table that volume and ACV of optimal bicluster (i.e. largest bicluster in the population) of each dataset obtained using DFA is better than BPSO.



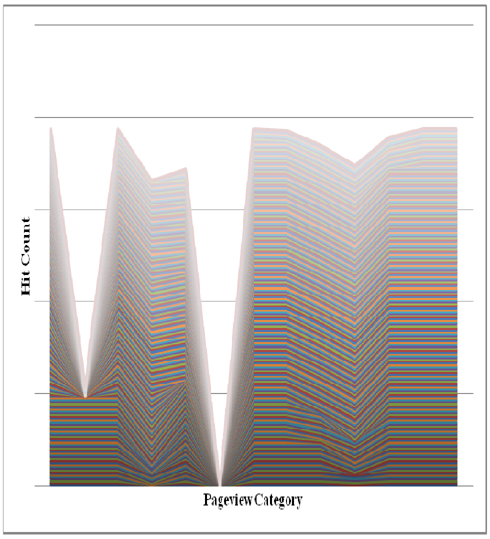
(a)



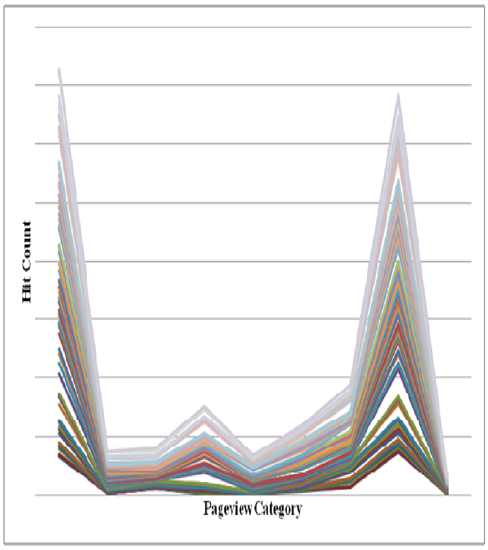
(b)

Fig. 2 Usage Pattern of Sizable Bicluster for MSWEB (a) and MSNBC Dataset (b) using DFA

In Figs. 2 and 3, X-axis represents the pageview categories in the optimal bicluster extracted from each dataset viz., MSWEB and MSNBC. Y-axis represents the hit count value of each pageview category in the bicluster. Figs. 1 (a) and (b) show the usage pattern of the optimal bicluster (i.e., sizable bicluster) extracted from the MSWEB dataset and MSNBC dataset using DFA based biclustering algorithm. Figs. 2 (a) and (b) show the usage pattern of the optimal bicluster (i.e., sizable bicluster) extracted from the MSWEB dataset and MSNBC dataset using BPSO based biclustering algorithm. It is evident from Figs. 1 and 2 that, DFA extracts the sizable bicluster (high volume bicluster) and high ACV bicluster without trapping at local optimal.



(a)



(b)

Fig. 3 Usage Pattern of Sizable Bicluster for MSWEB (a) and MSNBC Dataset (b) using BPSO

TABLE I
COMPARATIVE RESULTS FOR MSNBC DATASET

Algorithm	DFA	BPSO
Mean ACV	0.9742	0.9547
Mean Volume	8902.6	7823.5
Degree of Overlap	0.0012	0.0047
User Coverage (%)	89.36	82.04
Page Coverage (%)	90.2	81.04

TABLE II
COMPARATIVE RESULTS FOR MSWEB DATASET

Algorithm	DFA	BPSO
Mean ACV	0.9842	0.942
Mean Volume	6902.6	6150.3
Degree of Overlap	0.0035	0.0107
User Coverage (%)	92.3	81.59
Page Coverage (%)	89.7	79.5

TABLE III
CHARACTERISTIC OF OPTIMAL BICLUSTER (I.E., LARGEST BICLUSTER)

Algorithm	DFA		BPSO	
	MSWEB	MSNBC	MSWEB	MSNBC
ACV	0.9903	0.99	0.9713	0.9887
Volume	11676	9537	9650	8019
No. of Users	973	867	965	891
No. of Pages	12	11	10	9

VI. CONCLUSION

In this paper, a preliminary empirical evaluation of the performance of a new bio-inspired metaheuristic, Discrete Firefly Algorithm while attempting the non-trivial biclustering task. The results achieved for two real world datasets suggest that in terms of size and ACV value, the bicluster obtained in DFA method are far better than the biclusters obtained in BPSO algorithm. This optimal bicluster is a representation of a set of correlated users which appears to have similar web usage behaviour under subset of pages of a web site. Hence it can be aggregated to represent the optimal usage profile (i.e., a collection of web pages or URLs) for a web site. Usage profile discovered is effective in capturing user-to-page relationship and similarities at the level of user sessions. This knowledge can be used in the applications like personalization, target marketing etc. Future works will focus on some improvements for the proposed DFA based biclustering algorithm with regard to the locating largest bicluster with high degree of coherence among users and to the fitness function.

REFERENCES

[1] Cheng, Y., Church, G.M.: Biclustering of Expression Data. In: International Conference on Intelligent Systems for Molecular Biology, 2000, pp. 93–103.
[2] Coelho, G.P., de França, F.O., Von Zuben, F.J.: Multi-Objective Biclustering: When Nondominated Solutions are not Enough. J. Math. Model Algor. Vol. 8, 2009, pp:175–202.
[3] Das C, Maji P , Chattopadhyay S, A Novel Biclustering Algorithm for Discovering Value-Coherent Overlapping σ-Biclusters, Advanced Computing and Communications, 2008, pp:148-156,.
[4] David Beasley, David R. Bull, and Ralph R. Martin, An overview of genetic algorithms: Part 2, research topics. University Computing, Vol. 15, No. 4,1993, pp: 170–181.
[5] de Castro, P.A.D., de França, F.O., Ferreira, H.M., Von Zuben, F.J., Applying Biclustering to Perform Collaborative Filtering. In: International Conference on Intelligent System Design and Applications, 2007, pp. 421–426.
[6] de Castro, P.A.D., de França, F.O., Ferreira, H.M., Von Zuben, F.J. , Applying Biclustering to Text Mining: An Immune-Inspired Approach. In: de Castro, L.N., Von Zuben, F.J., Knidel, H. (eds.) ICARIS 2007. LNCS, vol. 4628, pp. 83–94. Springer, Heidelberg, 2007.
[7] de França, F.O., Coelho, G.P., Von Zuben, F.J. , bicACO: An Ant Colony Inspired
[8] Divina, F., Aguilar-Ruiz, J.S.: Biclustering of Expression Data with Evolutionary Computation. IEEE Trans. Knowl. Data Eng. Vol. 18, 2006, pp:590–602.
[9] Kennedy, J. and Eberhart, R.C., A discrete binary version of the particle swarm algorithm, Systems, Man, and Cybernetics, 1997. 'Computational Cybernetics and Simulation,IEEE International Conference ,vol.5,pp. :4104 - 4108 ,1997.
[10] Madeira, S.C., Oliveira, A.L.: Biclustering Algorithms for Biological Data Analysis: A Survey. IEEE/ACM Trans. Comput. Biol. Bioinform. Vol.1,, 2004, pp:24–45.
[11] Mitra, S., Banka, H. ,Multi-objective Evolutionary Biclustering of Gene Expression Data. Pattern Recogn. Vol. 39, 2006, pp: 2464–2477.
[12] Mobasher, B., Dai, H., Nakagawa, M., Luo, T., Discovery and evaluation of aggregate usage profiles for web personalization. Data Mining and Knowledge Discovery, Vol.6, 2002, pp. 61–82.
[13] R. Rathipriya, K. Thangavel, J. Bagyamani, Binary Particle Swarm Optimization based Biclustering of web Usage Data, International Journal Computer Applications, Vol. 25 No.2, , 2011, pp:43-49.
[14] R. Rathipriya, K. Thangavel, J. Bagyamani, Evolutionary Biclustering of Clickstream Data ,International Journal of Computer Science Issues, Vol. 8, 2011, pp:32-38.
[15] Smith, K.A. and A. Ng, Web Page Clustering using A Self-Organizing Map of User Navigation Patterns. Decision Support Systems, Vol. 35, 2003, pp : 245- 256.
[16] Srivastava, J., Cooley R., Deshpande, M., Tan, P.N., Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. SIGKDD Explorations, Vol. 1, No. 2, 2000, pp:12-23.
[17] Tanay, A., Sharan, R., Shamir, R.: Biclustering Algorithms: A Survey. In: Srinivas, A. (ed.) Handbook of Computational Molecular Biology, Chapman & Hall/CRC , 2005.
[18] Teng L, Chan L, Discovering biclusters by iteratively sorting with weighted correlation coefficient in gene expression data., J Signal Process Syst, Vol. 50, No. 3, 2008, pp:267-280.
[19] Xie, B., Chen, S., Liu, F.: Biclustering of Gene Expression Data Using PSO-GA Hybrid. In: International Conference Bioinformatics and Biomedical Engineering, pp. 302–305, 2007.
[20] Xu . R, Wunsch. D ,Survey of clustering algorithms, IEEE trans. on Neural Networks, vol. 16, No. 3, 2005, pp: 645-678.
[21] Yang, X-S., Firefly algorithms for multimodal optimization, in: Stochastic Algorithms: Foundations and Applications, SAGA, Lecture Notes in Computer Sciences, Vol. 5792, 2009, pp:169-178.

Thangavel K, is presently the Prof. & Head, Department of Computer Science Periyar University, Salem. He has completed his PhD in Gandhigram Rural University. His areas of interest are data mining, image processing, mobile computing and rough set theory and Optimization Techniques. He is reviewer of reputable journals. He has received Young Scientist Award 2009 from Tamilnadu State Council for Science and Technology.

Rathipriya R, is working as Assistant Professor in Periyar University, Salem, India. She received her Bachelor of Science and Master of Science degrees in Computer Science from the Periyar University. She has completed PhD in Bharathiyar University, India. Her research interests are in several areas of data mining, web mining, Optimization techniques and Bio-Informatics.