World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:8, No:2, 2014

Comparative Study on Swarm Intelligence Techniques for Biclustering of Microarray Gene Expression Data

R. Balamurugan, A. M. Natarajan, K. Premalatha

***Abstract*—**Microarray gene expression data play a vital in biological processes, gene regulation and disease mechanism. Biclustering in gene expression data is a subset of the genes indicating consistent patterns under the subset of the conditions. Finding a biclustering is an optimization problem. In recent years, swarm intelligence techniques are popular due to the fact that many real-world problems are increasingly large, complex and dynamic. By reasons of the size and complexity of the problems, it is necessary to find an optimization technique whose efficiency is measured by finding the near optimal solution within a reasonable amount of time. In this paper, the algorithmic concepts of the Particle Swarm Optimization (PSO), Shuffled Frog Leaping (SFL) and Cuckoo Search (CS) algorithms have been analyzed for the four benchmark gene expression dataset. The experiment results show that CS outperforms PSO and SFL for 3 datasets and SFL give better performance in one dataset. Also this work determines the biological relevance of the biclusters with Gene Ontology in terms of function, process and component.

***Keywords*—**Particle swarm optimization, Shuffled frog leaping, Cuckoo search, biclustering, gene expression data.

I. Introduction

DNA microarray technology measures the gene expression level of thousand of genes under multiple experimental conditions [1]. The conditions may belong to different time points or different environmental conditions. In a few cases the conditions may have come from cancerous tissues, healthy tissues, or different individuals. Later than 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. Every element of this matrix represents the expression level of a gene under a specific condition, and is represented by a real number. It is typically the logarithm of the relative plethora of the mRNA of the gene under the particular condition. Fig. 1 depicts the structure of 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 universal 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 instance, 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.

R. Balamurugan is with the Bannari Amman Institute of Technology, Erode, Tamil Nadu, India (e-mail:balacse05@gmail.com).

	Con 1	Con 2	Con 3	Con 4	Con 5
Gene 1	1.2	-3.4	2.6	1.9	-0.2
Gene 2	7.3	-2.1	-3.1	9.4	8.8
Gene 3	-0.7	0.9	2.1	-0.9	1.1
Gene 4	0.1	-0.2	4.5	-1.2	-3.9
Gene 5	-2.2	6.5	-1.3	-0.5	-6.9
Gene 6	3.8	-0.9	2.2	-0.1	-5.5

Fig. 1 Structure of gene expression matrix

An investigation via clustering makes several premises that may not be completely adequate in all places. First the clustering can be applied to either genes or conditions; it implicitly directs the analysis of a particular aspect of the system. Secondly, clustering algorithms generally seek a disjoint cover of the set of elements, requiring that no gene or condition belongs to more than one cluster.

The idea of a bicluster arises to a more flexible computational framework. For instance 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 co-occurring clustering of both rows and columns of a gene expression data. Explicitness a bicluster is a sub matrix spanned by a set of genes and a set of conditions.

The difficulty 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 [2] given in:

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

Equation (2) approximates (1),

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

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

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

Discovering significant biclusters in an expression data is a more difficult problem than clustering [3] and it is an NP-hard problem [4]. The problem of finding a coherent bicluster 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 an individual solution. The 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 | $\forall$s $\in$ S; f(m) < f(s).

This work develops and implements the biclustering based on the most popular and robust bio inspired strategies Particle swarm optimization, Shuffled frog leaping and Cuckoo search algorithms. The remainder of this paper is organized as follows: Section II provides the structure of bicluster and related works in biclustering. Section III gives a general overview of the Particle swarm optimization, Shuffled frog leaping and Cuckoo search algorithms. Section IV presents the detailed experimental setup and results for comparing the performance of the PSO, SFL and CS. Section V gives the biological relevance of the biclusters with Gene Ontology for cuckoo search.

II. Bicluster

A. Structure of Biclusters

Madeira and Oliveira [5] have identified four major groups of structures inside the submatrices are,

a. Bicluster with constant value

$a_{ij} = \mu$

b. Bicluster with constant values on rows or columns

$(\ a_{ij} = \mu + \alpha_i$ or $a_{ij} = \mu * \alpha_i\ )$ and

$(\ a_{ij} = \mu + \beta_j$ or $a_{ij} = \mu * \beta_j\ )$

c. Bicluster with coherent values

$a_{ij} = \mu + \alpha_i + \beta_j$ or $a_{ij} = \mu * \alpha_i * \beta_j$

d. Bicluster with coherent evolutions

$a_{ih} \leq a_{ir} \leq a_{it} \leq a_{id}$ or $a_{hj} \leq a_{rj} \leq a_{tj} \leq a_{dj}$

The bicluster sets are classified according to their relative structure [22]. According to the particular properties of each problem, one or more of these various types of biclusters are generally considered interesting.

B. Review of Related Works

At very first the biclustering approach for gene expression data using Mean Square Residue (MSR) is proposed by [6]. Their algorithm adopts a sequential covering strategy in order to return a list of n biclusters from an expression data matrix. Statistical-Algorithmic Method for Bicluster Analysis (SAMBA), a biclustering algorithm that performs the simultaneous bicluster identification by using exhaustive enumeration [4]. Murali and Kasif intended 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 [7]. Ben-Dor et al. defined a bicluster as an Order-Preserving Sub-Matrix (OPSM) [8]. An Iterative Signature Algorithm (ISA) and provides a definition of biclusters as transcription modules to be retrieved from the expression data proposed by [9].

A Multi-Objective Evolutionary Algorithm (MOEA) based on Pareto dominancy presented by [10]. A Sequential Evolutionary BIclustering (SEBI) approach presented by [3].The term sequential refers the way in which bicluster are discovered, only one bicluster obtained per each run of the evolutionary algorithm. Liu & Wang introduced Maximum Similarity Bicluster (MSB) algorithm [11]. 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 is proposed by [12]. Liu et al. 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 [13]. Coelho et al. presented an immune-inspired algorithm for biclustering based on the concepts of clonal selection and immune network theories adopted in the original aiNet algorithm [14].

A Pattern-Driven Neighbourhood Search (PDNS) approach for the biclustering problem is proposed by [15]. Huang et al. proposed a new biclustering algorithm based on the use of an Evolutionary Approach (EA) together with hierarchical clustering [16]. Painsky and Rosset proposed Exclusive Row Biclustering for Gene Expression Using a Combinatorial Auction Approach [17]. It extracts the exclusive row biclusters via a combination of existing biclustering methods and combinatorial auction techniques. Ray et al. introduced a CoBi: Pattern Based Co-Regulated Biclustering of gene expression Data [18]. It is mainly used for grouping both positively and negatively regulated genes from microarray expression data. Recently a new biclustering algorithm based on association rule mining proposed by [19]. It grounded on association rule mining, which can support different well-known biclustering models.

III. Literature Review on Optimization Techniques

At present, there are so many optimization techniques available. This section presents a literature review of a few optimization techniques. The familiar methods linear programming [20], the quadratic programming [21] the dynamic programming [22] the Simplex method [23] and the gradient methods [24] are deterministic methods which make possible to resolve some types of optimization problems in a finished time period. In observation these problems are too complex and require too much time to resolve by deterministic methods. Metaheuristics are stochastic optimization it finds a solution in a reasonable time. Metaheuristics have usually an iterative behavior. The same pattern is repeated until a stopping criterion is met at the beginning for optimization.

A. Particulate Swarm Optimization

The particulate swarm optimization is a metaheuristic algorithm proposed by Kennedy and Eberhart [25]. This method inspired from animals social behavior in their moving in swarms. The most used example is the behavior of fish school [26], [27]. Indeed, these animals are characterized by a movement dynamics relatively complex, while individually each one has a limited intelligence and local knowledge focused on its position in the swarm. Thereby, each individual has knowledge only of the position and speed of its nearest neighbors. It therefore uses not only its own memory, but also local information of nearest neighbors to decide its own movement. Simple rules, such as "go with the same speed as others", "moving in the same direction" or "stay close neighbors" are among key behaviors that maintain cohesion of the swarm and allow the implementation of complex and adaptive collective behaviors. The swarm's global intelligence is a direct consequence of local interactions between different particles. Therefore, system performance as a whole is greater than the performance sum of its different parts.

Procedure for PSO:

1. Initialize a population of particles with random positions and velocities on N dimensions in the problem space.
2. For each particle, evaluate the desired optimization fitness function in N variables. Compare particle's fitness evaluation with its pbest. If current value is better than pbest, then set pbest equal to the current value, and P_i equals to the current location X_i in N-dimensional space.
3. Identify the particle in the swarm with the best success so far, and assign its index to the variable g. Change the velocity and position of the particle according to equations,

$$p_{id} = w \times v_{id} + c_1 \times rand() \times (p_{id} - x_{id}) + c_2 \times rand() \times (p_{gd} - x_{id})$$

$$x_{id} = x_{id} + v_{id}$$

4. Loop to step 2 until a criterion is met, typically a sufficiently good fitness or a maximum number of iterations.

B. Shuffled Frog-Leaping

The shuffled frog-leaping algorithm is a memetic metaheuristic that is designed to seek a global optimal solution [28]. It is based on the evolution of memes carried by individuals and a global exchange of information among the population [29]. In sprit, it combines the reimbursement of the local search method of the particle swarm optimization [25], and the core idea of mixing information from parallel local searches to move toward a global solution [30].

Procedure for SFL:

1. Initialize population of P frogs is created randomly.
2. For each frog, evaluate the desired optimization fitness function in P frogs.
3. Sort the frogs in a descending order according to their fitness.
4. Divide the entire population into m memeplexes, each containing n frogs. In this process, the first frog goes to the first memeplex, the second frog goes to the second memeplex, frog m goes to the m^{th} memeplex, and frog m+1 goes back to the first memeplex and so on. Within each memeplex, the frogs with the best and the worst fitness are identified as x_b and x_w, respectively.
5. Identify the frog with the global best fitness is defined as x_g.
6. Change the position of the frog with the worst fitness is adjusted as follows;

$$D_i = rand() \times (x_b - x_g)$$

$$x_{i+1} = x_i \times D_i \text{ where } -D_{max} \le D_i \le D_{max}$$

where *rand* is a random number between 0 and 1, and D_{max} is the maximum allowed change in a frog's position. If this process produces a better solution, it is replaced for the worst frog. Otherwise, the calculations in step 6 are repeated but with respect to the global best frog (i.e. X_b is replaced by X_g). If no improvement is possible, then a new solution is randomly generated to replace the worst frog. Hence, the calculations continue for a specific number of iterations.

C. Cuckoo Search

The cuckoo search is an optimization [31] based on the brood parasitism of the cuckoo species by laying their eggs in the nests of other host birds. 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. Each egg in a nest represents a solution, and a cuckoo egg represents a new solution. The better new solution (cuckoo) is replaced with a solution which is not so good in the nest.

Pseudo code for Cuckoo search:

1. Generate an initial population of n host nests;
2. while (t<Max Generation) or (stop criterion)
3. Get a cuckoo randomly (say, i) and replace its solution by performing Levy flights;
4. Evaluate its fitness Fi
5. Choose a nest among n (say, j) randomly;

6. if (Fi < Fj)
7. Replace j by the new solution;
8. end if
9. A fraction (pa) of the worse nests is abandoned and new ones are built;
10. Keep the best solutions/nests;
11. Rank the solutions/nests and find the current best;
12. Pass the current best to the next generation;
13. end while

D. Biclustering Representation

Each bicluster is encoded as an individual of the population (particle, frog, egg). Each population 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 a population, with a bit string for genes attached with another bit string for conditions. The population represents a candidate solution for this optimal bicluster generation problem. A bit is set to one if the corresponding gene and/or condition are present in the bicluster, and reset to zero otherwise. Fig. 2 shows an encoded representation of a bicluster.

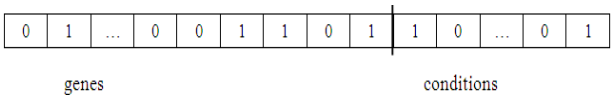

Fig. 2 Encoding representation of a bicluster

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 a population into a binary string representation of a bicluster is given in (4) as follows:

$$y_{ij} = \begin{cases} x_{ij} < 0.5 & 0 \\ otherwise & 1 \end{cases} \quad (4)$$

where

x_i - Random value generated for j^{th} gene/condition of i^{th} population

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. Fig. 3 shows the representation of the solution and its mapped bicluster representation.

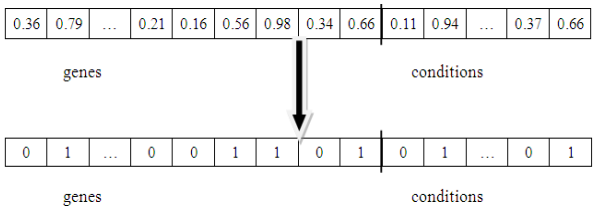

Fig. 3 Representation of the solution and its mapping to bicluster

E. Fitness Function

Mean Squared Residue problem has been introduced by Cheng and Church [6] 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, $R = \{r_1, r_2, ..., r_M\}$ and its set of columns $C = \{c_1, c_2, ..., c_N\}$. 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 [6] 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 aij 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 (5)

$$H(I,J) = \frac{1}{|I||J|} \sum_{i \in I} \sum_{j \in J} r_{i,j}^2 \quad (5)$$

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 gene variance can be represented in (6) as follows:

$$Var_r(I, J) = \frac{1}{|I|}\sum_{i \in I} v_r(i)$$
$$v_r(i) = \frac{1}{|J|}\sum_{j \in J}\left(a_{i,j} - a_{i,J}\right)^2 \quad (6)$$

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 (7) as follows:

$$f(I, J) = H(I, J) + \frac{1}{Var(I, J)} \quad (7)$$

to be derogated. In this way, the smaller the residue and the larger the gene variance are, the smaller the fitness value, i.e., the better the quality of that bicluster is.

IV. Experimental Analysis

A. Data sets

We have implemented the biclustering algorithm on four micro array data sets. In order to study its performance, namely the yeast *Saccharomyces cerevisiae* stress expression data [32], *Arabidopsis thaliana* expression data [33], yeast *Saccharomyces cerevisiae* cell cycle expression data [34] and rat CNS expression data [35] are used. The first Gasch yeast is *Saccharomyces cerevisiae* with 2993 genes and 173 conditions. The second one Arabidopsis thaliana expression data contain 734 genes and 69 conditions. The third dataset yeast *Saccharomyces cerevisiae* cell cycle expression contains 2884 genes and 17 experimental conditions. The rat CNS dataset has set of 112 genes under 9 conditions. The setting values of algorithmic control parameters of the mentioned algorithms are given below:

- PSO Settings: $c_1 = c_2 = 1.80$ and $\omega = 0.60$ have been used as recommended in [23].
- SFL Settings: p=20 and m=20 has been used as recommended in [26].
- CS Settings: $\lambda = 1.50$ and $p_a = 0.25$ have been used as recommended in [28].

Figs. 4-7 show the fitness value obtained for yeast *Saccharomyces cerevisiae* stress expression data, *Arabidopsis thaliana* expression data, yeast *Saccharomyces cerevisiae* cell cycle expression data and rat CNS data respectively. The SFL performs better on yeast stress expression data and remaining three datasets CS outperforms PSO & SFL. The solution update in PSO is obtained through the personal best and global best of the particle position. It causes premature convergence.

Because of the update is done within the population. The SFL contains number of memeplexes and the frogs in a memeplexes are shuffled after a designated number of iteration. It is a basis for diversification. In CS the fraction of worst solutions are destroyed and new solutions are generated periodically. So always there has been a diversification in solution. Clearly Fig. 8 shows the sample bicluster of size 16 × 4 for rat CNS data.

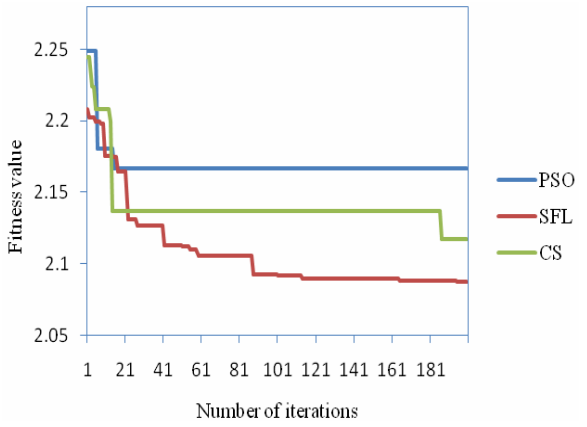

Fig. 4 Fitness value obtained for yeast stress data

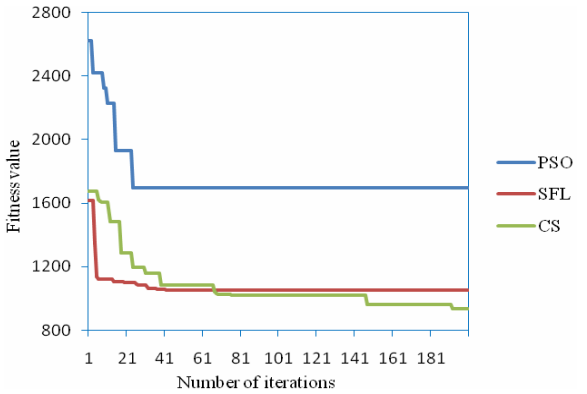

Fig. 5 Fitness value obtained for arabidopsis thaliana data

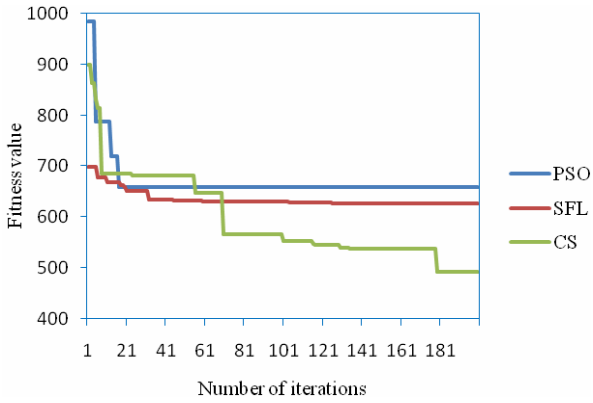

Fig. 6 Fitness value obtained for yeast cell data

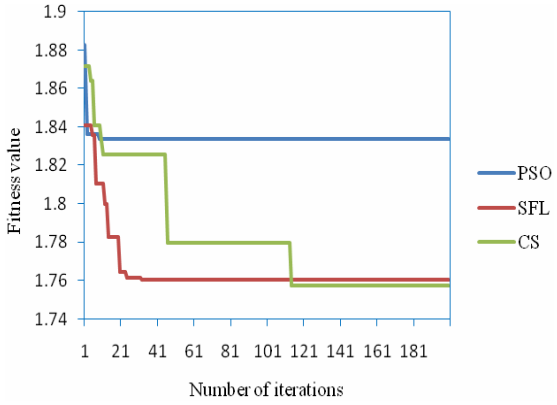

Fig. 7 Fitness value obtained for rat CNS data

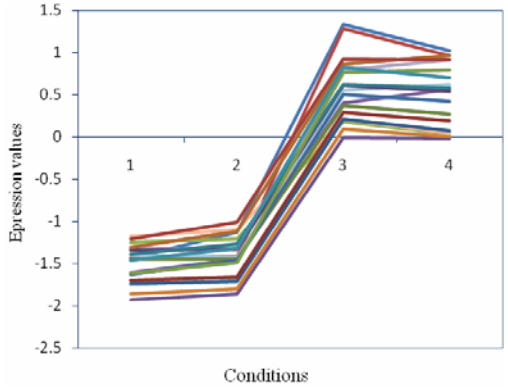

Fig. 8 Sample biclusters of size 16 × 4 for rat CNS data

V. Biological Analysis of Biclusters

The cuckoo search finds the biological relevance of the biclusters on the yeast *Saccharomyces cerevisiae* cell cycle expression data in terms of the statistically significant GO annotation database. The Gene Ontology (GO) project provides three structured, controlled vocabularies that describe gene products in terms of their associated biological processes, cellular components and molecular functions in a species independent manner. The mining results to be understood by feeding the genes in each bicluster to Onto-Express and a hierarchy of functional annotations is obtained in terms of Gene Ontology for each bicluster. The degree of enrichment is calculated by p-values which use a cumulative hyper geometric distribution to compute the probability of observing the number of genes from a particular GO category (function, process and component) within each bicluster. The p-value is the probability that the genes are selected into the cluster by random. A small p-value implies that the cluster is highly differed found by chance. The annotations of genes for five ontologies including biolgoical process, cellular component, molecular function, deletion viability and regulatory pathway are obtained.

A. Biological Annotation for Yeast Cell Cycle Using GOTermFinder Toolbox

In order to discover the biological annotations for the biclusters, we use the GOTermFinder which is tool available in the Saccharomyces Genome Database (SGD) [36]. GOTermFinder is deliberate to search for the significant shared GO terms of the groups of genes and provides users with the means to identify the characteristics that the genes may have in common. Table I lists the significant common GO terms (or parent of GO terms) used to describe the set of genes in each bicluster for the process, function and component ontologies. Only the most significant terms are shown. For example to the bicluster BC_1, the genes are mainly involved in DNA binding activity. The tuple (n=473, $p=6.87\times10^{-6}$) represents that out of 1482 genes in bicluster BC_1, 473 genes belong to binding activity function, and the statistical significance is given by the p-value of $p=6.87\times10^{-6}$.

TABLE I
Significant GO Terms (Process, Function, Component) for Three Biclusters on Yeast Cell Cycle Data

Bic. No.	No. of. Genes	Process	Function	Component
BC_1	1482	single organism (n=1021, $p=5.97\times10^{-86}$)	DNA binding (n=473, $p=6.87\times10^{-6}$)	intracellular organelle (n=1152, $p=1.72\times10^{-72}$)
BC_7	1500	cellular (n=1384, $p=6.27\times10^{-162}$)	hydrolase (n=296, $p=4.12\times10^{-16}$)	cell part (n=1434, $p=1.09\times10^{-98}$)
BC_{12}	1491	metabolic (n=1127, $p=2.92\times10^{-93}$)	transferase (n=238, $p=5.14\times10^{-12}$)	intracellular part (n=1354, $p=5.30\times10^{-83}$)

V. Conclusion

Swarm intelligence techniques are based on collective intelligence of groups of simple agents. All the swarm intelligence techniques are not efficient for all real time problems. A few algorithms are very efficient and they are popular tools for solving real-world problems. In this work the popular swarm intelligence techniques such as PSO, SFL and CS is applied for biclustering micro array gene expression data. Biclustering finds subsets of genes that show similar patterns under a specific subset of experimental conditions. The experimental results are analyzed with 4 different benchmark data sets. The results show that CS outperforms 3 data sets and SFL for one data set.

Acknowledgment

This work is supported by the research grant from Department of Biotechnology (DBT), Government of India, New Delhi, Grant No. BT/PR3741/BID/382/2011.

References

[1] D.J. Lockhart, and E.A. Winzeler, "Genomics, gene expression and DNA arrays," *Nature*, vol. 405, pp. 827-836, June 2000.
[2] C.L. Liu, *Introduction to Combinatorial Mathematics*. New York, McGraw-Hill Publication, 1968.
[3] F. Divina, and J.S. Aguilar-Ruiz, "Biclustering of expression data with evolutionary computation," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 5, pp. 590-602, May 2006.
[4] A. Tanay, R. Sharan, and R. Shamir, "Discovering statistically significant biclusters in gene expression data," *BMC bioinf.*, vol. 18, pp. 136-144, March 2002.
[5] S.C. Madeira, and A.L. Oliveira, "Biclustering algorithms for biological data analysis: A survey," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 1, no. 1, pp. 24-45, Jan. 2004.

[6] Y. Cheng, and G.M. Church, "Biclustering of Expression Data," in *Proc. of the 8th Conf. Intel. Sys. Mol. Biol.*, Menlo Park, United States, 2000, pp. 93-103.
[7] T. Murali, and S. Kasif, "Extracting conserved gene expression motifs from gene expression data," in *Pacific Symposium on Bio computing,* Boston University, United States, 2003, pp. 77-88.
[8] A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini, "Discovering local structure in gene expression data: The order-preserving submatrix problem," *J. Comput. Biol.*, vol. 10, no. 4, pp. 373-384, 2003.
[9] S. Bergmann, J. Ihmels, and N. Barkai. "Iterative signature algorithm for the analysis of large-scale gene expression data," *Phys. Rev. E*, vol. 67, pp. 1-18, March 2003.
[10] S. Mitra, and H. Banka. "Multi-objective evolutionary biclustering of gene expression data," *Pattern Recognit. Lett.*, vol. 39, no. 12, pp. 2464-2477, Dec. 2006.
[11] X. Liu, and L. Wang, "Computing the maximum similarity bi-clusters of gene expression data," *BMC bioinf.*, vol. 23, no. 1, pp. 50-56, Oct. 2007.
[12] P. DiMaggio, S. McAllister, C. Floudas, X. Feng, J. Rabinowitz, and H. Rabitz, "Biclustering via optimal re-ordering of data matrices in systems biology: rigorous methods and comparative studies," *BMC bioinf.*, vol. 9, no. 1, pp. 458-467, Oct. 2008.
[13] J. Liu, Z. Li, X. Hu, and Y. Chen, "Biclustering of microarray data with mospo based on crowding distance," *BMC bioinf.*, vol. 10, no. 9, April 2009.
[14] G.P. Coelho, F.O. de Franca, and F.J.V. Zuben, "Multi-objective biclustering: When non-dominated solutions are not enough," *J. Math. Model. Algorithms*, vol. 8, no. 2, pp. 175-202, June 2009.
[15] W. Ayadi, M. Elloumi, and J. Hao, "Pattern-driven neighborhood search for biclustering microarray data," *BMC bioinf.*, vol. 13, no. 7, May, 2012.
[16] Q. Huang, D. Tao, X. Li, and A.W.C. Liew, "Parallelized evolutionary learning for detection of biclusters in gene expression data," *IEEE/ACM Trans.Comput. Biol. and Bioinf.*, vol. 9, no. 1, pp. 560-570, April 2012.
[17] A. Painsky, and S. Rosset, "Exclusive Row Biclustering for Gene Expression Using a Combinatorial Auction Approach," in *Proc. IEEE 12th Inter. Conf. Data Mining*, Belgium, pp.1056-1061, Dec. 2012.
[18] S. Roy, D.K. Bhattacharyya, and J.K. Kalita, "CoBi: Pattern Based Co-Regulated Biclustering of Gene Expression Data," *Pattern Recognit. Lett.*, vol. 34, no. 14, pp. 1669-1678, Oct. 2013.
[19] B. Boutsinas, "A New Biclustering Algorithm Based On Association Rule Mining," *Int. J. Artif. Intell. Tools,* vol. 30, no. 3, 2013
[20] A. Schrijver, *Theory of Linear and Integer Programming*, New York, John Wiley & Sons, 1998.
[21] J. Nocedal, and S. J. Wright, *Numerical Optimization*, Springer Series in Operations Research. Springer, 1999.
[22] D.P. Bertsekas, *Dynamic Programming and Optimal Control*. 2nd ed. Athena Scientific, Belmont, Massachusetts, USA, 2000.
[23] J.A. Nelder, and R. Mead, "A simplex method for function minimization," *Comput. J.*, vol. 7, no. 2, pp. 308-313, 1965.
[24] M. Avriel, *Nonlinear Programming: Analysis and Methods*, United Kingdom, Dover Publishing, 2003.
[25] J. Kennedy, and R.C. Eberhart, Particle swarm optimization. in *Proc. IEEE Inter. Conf. Neural Networks*, Piscataway, NJ, United states, pp. 1942-1948, 1995.
[26] C.W. Reynolds, "Flocks, herbs, and schools: A distributed behavioral model," in *Proc. 14th annu. conf. Comput. graphics and interactive techniques,* New York, vol. 21, July 1987, pp. 25-34.
[27] E.O. Wilson, *Sociobiology: The new synthesis*, Massachusetts, Belknap Press, Cambridge, 1975.
[28] M.M. Eusuff, K. Lansey, and F.Pasha, "Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization," *Eng. Optim.*, vol. 38, no. 2, pp. 129-154, Jan. 2006.
[29] M.M. Eusuf, K.E. Lansey, "Optimization of water distribution network design using the shuffled frog leaping algorithm," *J. Water Resour. Plann. Manage*, vol. 129, no. 3, pp. 210-225, May 2003.
[30] Q.Y. Duan, V. K. Gupta, and S. Sorooshian, "Shuffled complex evolution approach for effective and efficient global minimization," *J. Optim. Theory Appl.,* vol. 76, pp. 502-521, March 1993.
[31] X.S. Yang, and S. Deb, "Engineering optimisation by Cuckoo search" *Int. J. Math. Modeil. Numer. optim.*, vol. 1, no. 4, pp. 330-343, Dec. 2010.
[32] A.P. Gasch, P.T. Spellman, C.M. Kao, O. Carmel Harel, M.B. Eisen, G. Storz, D. Botstein, and P.O. Brown, "Genomic Expression Programs in the Response of Yeast Cells to Environmental Changes," *Mol. Biol. Cell.*, vol. 11, no. 12, pp. 4241-4257, Oct. 2000.
[33] S. Bleuler, A. Prelic, and E. Zitzler, "An EA framework for biclustering of gene expression data," in *Congress of IEEE on Evolutionary Comput.*, Switzerland, vol. 1, 2004, pp. 166-173.
[34] R. Cho, M.J. Campbell, A.E. Winzeler, L. Steinmetz, and A. Conway, "Genome-Wide Transcriptional Analysis of the Mitotic Cell Cycle," *Mol. Cell*, vol. 2, no. 1, pp. 65-73, July 1998.
[35] X. Wen, S. Fuhrman, G.S. Michaels, D.B. Carr, S. Smith, J.L. Barker, and R. Somogyi, "Large-scale temporal gene expression mapping of central nervous system development," *in Proc. National Academy of sciences*, vol. 95, no.1, pp. 334-339, Nov. 1998.
[36] Saccharomyces Genome Database Source: http://www.yeastgenome.org/cgibin/GO/goTermFinder.pl.