2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Biclustering of Gene Expression Data Using Particle Swarm Optimization Integrated with Pattern-Driven Local Search

Yangyang Li, Member, IEEE, Xiaolong Tian, Licheng Jiao, Senior Member, IEEE and Xiangrong Zhang, *Member, IEEE*

Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, International Research Center for Intelligent Perception and Computation, Xidian University, Xi'an 710071, China
yyli@xidian.edu.cn

***Abstract*-Biclustering is of great significance in the analysis of gene expression data and is proven to be a NP-hard problem. Among the existing intelligent optimization algorithms used in the gene expression data analysis, most concentrate on the global search ability but ignore the inherent trajectory information of gene expression data, so the search efficiency is low. In this paper, a pattern-driven local search operator is incorporated in the binary Particle Swarm Optimization (PSO) algorithm in order to improve the search efficiency. Experiments show that our approach is valid.**

***K*eywords-Biclustering, Gene expression data, Particle swarm optimization (PSO), Pattern-driven.**

I. INTRODUCTION

Microarray technology which plays a great role in biological research makes a great development in recent years and has produced a mass of gene expression data to be analyzed. Gene expression data are typically represented by a data matrix where each row contains the gene expression values under different conditions. Clustering is a commonly used technique in the analysis of gene expression data, but traditional clustering techniques are based on the whole conditions of each gene, they cannot cluster genes into different groups under different subsets of conditions which is essential with respect to a cellular process. In the cellular process, different gene subsets behave almost independently under different condition subsets. As a result, traditional clustering techniques are no longer suitable for the analysis of gene expression data. Biclustering techniques can overcome the disadvantage of traditional clustering techniques.

Biclustering simultaneously clusters the rows (genes) and columns (conditions) in order to find subsets of genes which behave similarly under the specified subsets of conditions. The concept of biclustering was first introduced into the field of gene expression data analysis in 2000 by Cheng and Church, Cheng and Church algorithm [1] is a greedy iterative search algorithm which iteratively adds and removes rows and columns according to the evaluation score of a bicluster measured by the Mean Squared Residues (MSR). The MSR is introduced in Section III.

In recent years, several algorithms about biclustering have been proposed, some examples can be found in [2]-[4]. These methods can be roughly divided into the following categories [11]: greedy iterative search, biclusters enumeration and stochastic search. Recently some excellent intelligent optimization algorithms have been used to solve the biclustering problem, such as [5] [6].

Most of the existing intelligent optimization algorithms used in biclustering concentrate on the ability to search over the whole possible space without using the inherent trajectory information of gene expression data. In this paper, a pattern-driven local search operator is integrated into the binary PSO algorithms to improve the search efficiency.

The rest of this paper is organized as follows: Section II gives a detailed introduction of gene expression data. In section III, the proposed algorithm is described in detail. Section IV reveals our experiment results. Finally, the last section is used to conclude our work.

II. GENE EXPRESSION DATA

Gene expression data reflect the enrichment of the message Ribonucleic Acid (mRNA). The data are mainly obtained by two high flux detection technology, the complementary Deoxyribonucleic Acid (cDNA) microarrays and Oligonucleotide microarrays. These two techniques can be used to analyze the correlation of different genes, how the activity of a gene is affected under different conditions, and the change of the same gene under different conditions. Different conditions are taken into consideration, including different tissues, the various stages of the cell cycle, drug action time, and different patients.

The analysis of gene expression data is significant, through which some useful information can be extracted to apply in medical and biological studies. Biclustering techniques aim at extracting gene sets with similar patterns under different condition subsets.

Cheng and Church [1] first introduced the concept of biclustering into the field of the gene expression data analysis, and Madeira and Oliveira [7] give a formal definition of biclustering. Gene expression data are represented by a data

This work was supported by the Program for New Century Excellent Talents in University (No. NCET-12-0920), the National Natural Science Foundation of China (Nos. 61272279, 61272282, 61371201, and 61203303), the Fundamental Research Funds for the Central Universities (Nos. K5051302049, K5051302023, K50511020011, K5051302002 and K5051302028) and the Fund for Foreign Scholars in University Research and Teaching Programs (the 111 Project) (No. B07048).

matrix M where $m_{i,j}$ represents the expression level of gene i under condition j. A bicluster can be represented as $M_{I,J}$ where $I \subseteq R$ and $J \subseteq C$, R and C are the sets of all the row index sets and column index sets respectively.

III. Our Proposed Algorithm

A. Simple Binary PSO

The PSO algorithm was proposed by Eberhart and Kennedy in 1995 [8]. PSO is a population-based intelligent search progress, each member of the population is called particle, a potential solution for the problem to be optimized. It simulates the cluster behavior of insects, birds and fish, these groups find food in a cooperative way, each member of the groups keeps changing the path by learning the experience of itself and other members to find food.

PSO is mainly used for optimization problem in continuous space, which is characterized by fast convergence speed and is easy to understand and implement by programming. Kennedy J. and Eberhart R. C. proposed a binary PSO in order to solve some practical problems in discrete space in [9]. Li-Yeh Chuang *et al*. applied binary into feature selection with gene expression data [10], with which good experimental results were obtained.

Iterative formula of simple binary PSO [8] can be defined as:

$$v_n^{t+1} = w * v_n^t + c_1 * r_1 * \left(Pbest_n - x_n^t\right) + c_2 * r_2 * \left(Gbest - x_n^t\right) \quad (1)$$

$$x_n^{t+1} = x_n^t + v_n^{t+1} \quad (2)$$

where, v_n^t and v_n^{t+1} devote the velocity of the n^{th} particle in t^{th} generation and t^{t+1} generation relatively, w is inertia weight, c_1 and c_2 are learning factors, r_1 and r_2 are both random numbers between 0 and 1, $Pbest_n$ devotes the history optimal position of the n^{th} particle, $Gbest$ devotes the current optimal position of the whole particle swarm.

Its binary version[10] can be given as:

$$v_n^{t+1} = w * v_n^t + c_1 * r_1 * \left(Pbest_n - x_n^t\right) + c_2 * r_2 * \left(Gbest - x_n^t\right) \quad (3)$$

$$s\left(v_{i,j}\right) = \frac{1}{1+e^{-v_{i,j}}} \quad (4)$$

$$\text{if}(\text{rand} < s(v_{i,j})) \text{ then } x_{i,j} = 1; \text{else } x_{i,j} = 0 \quad (5)$$

where $s()$ is sigmoid function, $v_{i,j}$ devotes the velocity of the i^{th} particle in the j^{th} dimension, $x_{i,j}$ devotes the position of the i^{th} particle in the j^{th} dimension.

The values of δ, c_1, c_2 have something to do with the performance of the PSO algorithm, but we do not concentrate on these factors, some empirical values are set to these parameters, seen in section IV.

B. Pattern-driven Local Search Operator

Pattern-driven local search uses the trajectory of gene expression data to find its neighborhood. This operator is conducted on the neighborhood of a bicluster, before that the neighborhood must be determined. A behavior matrix M' needs to be computed from the original data matrix to find the neighborhood of a bicluster. Each row of M' represents the trajectory pattern of the genes across all the combined conditions while each column represents the trajectory pattern of all the genes under each pair of the selected conditions [11].

The behavior matrix is constructed as follows:

$$M'[i,l] = \begin{cases} 1, if & M[i,k] < M[i,q] \\ 0, if & M[i,k] = M[i,q] \\ -1, if & M[i,k] > M[i,q] \end{cases} \quad (6)$$

where $i \in [1..m], l \in [1..n'*(n'-1)/2], k \in [1, n'-1], q \in [2..n']$, $q \geq k+1$, m is the number of all the genes(rows), n' is the number of the selected conditions(columns).The value 1 indicates that the trajectory of gene i raise from k to q,-1 indicating falling, and 0 indicating no changing.

The behavior matrix gives some useful information about the trajectory of the genes across the selected conditions, by which the pattern of a bicluster is determined, and then the neighbors of the bicluster can be found. Fig.1 shows an example to determine the bicluster pattern.

A threshold α is defined in advance here to measure how one gene is similar to the determined bicluster pattern. If the matching rate of one particle to bicluster pattern is less than α, the gene is considered as a bad one, it will be removed so that some good genes can be selected from its neighborhood of the bicluster. The matching rate of a gene is defined as the ratio of the number of the same value within the corresponding trajectory in behavior matrix .The neighborhood can be determined according to the trajectories of all the genes across all the combinations of the selected conditions. If matching rate of one gene is greater than α, the gene is considered as one of the neighborhoods of the bicluster. If the number of the genes removed is greater than the number of neighbors, the neighbors will be added into the bicluster, otherwise, a row addition operator is needed to select good genes as many as the removed genes.

(a)

	c_1c_2	c_1c_3	c_1c_4	c_2c_3	c_2c_4	c_3c_4
g_1	1	0	1	-1	1	-1
g_2	0	1	1	0	-1	1
g_3	1	0	0	0	-1	1

(b)

|1|=2 |0|=1 |-1|=0	|1|=1 |0|=2 |-1|=0	|1|=2 |0|=1 |-1|=0	|1|=0 |0|=2 |-1|=1	|1|=1 |0|=0 |-1|=2	|1|=2 |0|=0 |-1|=1

(c)

1	0	1	0	-1	1

Fig.1 Example to determine bicluster pattern. (a) is the behavior matrix M' of a bicluster which contains three genes under four conditions, (b) is the extraction of dominating value of each column, and (c) is the pattern of the bicluster. |x| represents the number of the x over the corresponding column.

C. Fitness Function

Several evaluation functions can be used to evaluate the quality of a bicluster. One of the popular evaluation functions is the Mean Squared Residue(MSR) proposed by Cheng and Church [1], which has been used by several biclustering algorithms [12]-[14]. Though MSR has one disadvantage that it is unfit for detecting some models, such as shifting and scaling patterns [15]-[17], this paper still adopts the MSR as the evaluation function for that this paper pays more attention on the search efficiency of the PSO integrated with the pattern-driven local search operator. In this paper, the fitness function is designed to maximize the volumes as well as to minimize the MSR of a bicluster determined by each particle.

Fitness function is formally defined as follows:

$$fitness = \begin{cases} |I|*|J|, & if \quad MSR \le \delta \\ \dfrac{\delta}{MSR}, & otherwise. \end{cases} \tag{7}$$

where I and J devote the subsets of rows and subsets of columns respectively, δ is a threshold defined in advance.

The MSR can be computed as follows:

$$MSR = \frac{1}{|I|*|J|} \sum_{i \in I, j \in j} r(i,j)^2 \tag{8}$$

where $r(i,j) = m_{i,j} - m_{i,J} - m_{I,j} + m_{I,J}$

$$m_{I,J} = \frac{\sum_{i \in I, j \in J} m_{i,j}}{|I|*|J|}$$

$$m_{I,j} = \frac{\sum_{i \in I} m_{i,j}}{|I|}$$

$$m_{i,J} = \frac{\sum_{j \in J} m_{i,j}}{|J|}$$

A high value of MSR indicates that the bicluster is weakly similar while a low value indicates that the bicluster is strongly similar. The optimal object is to maximize the fitness.

D. Encoding

In this paper, a bicluster is encoded as a particle position represented by a fixed-size binary string with a part for genes and the other for conditions. A particle or a bicluster has the same meaning, which we use alternatively. This type of encoding is adopted by most of the existing evolutionary algorithms. If one gene or condition is included in a bicluster, the corresponding bit is set to 1, otherwise 0. Fig.2 shows an example for the encoding of a bicluster.

E. Pseudo-code of the Proposed Algorithm.

A binary PSO without local search operator is carried out as a framework, into which a pattern-driven local search operator will be integrated to show the search efficiency. Fig.3 gives the pseudo-code of the proposed algorithm.

The pseudo-code of the local search operator is given in Fig.4. The behavior matrix used in this operator is constructed according to (6). When a gene is selected from the neighbors, it is first evaluated by fitness function whether it can improve the bicluster determined by the particle.

Gene Expression Data M

	c_1	c_2	c_3	c_4	c_5
g1	22	38	42	44	23
g2	18	30	14	17	13
g3	29	52	43	39	41
g4	26	16	35	40	40
g5	5	9	-13	-14	-19

A Bicluster

1	2	3	4	5	1	2	3	4	5
0	1	1	0	1	1	0	1	0	1

Genes Conditions

Fig.2 Encoding of a bicluster. The dark cell is the corresponding submatrix determined by the bicluster.

IV. Experimental Results

This paper concentrates on the feasibility of the pattern-driven local search operator, so we only integrate the local search operator into a simple binary PSO algorithm. To investigate the performance of the novel-version PSO, particle swarm positions are initialized randomly instead of being initialized by some results obtained by other iterative greedy search approaches [18].

A. Experiment Environment and Datasets Used

The proposed algorithm is implemented in Matlab R2013. The real dataset Yeast Saccharomyces cerevisiae cell cycle expression dataset presented in [1] is investigated, it consists of expression values of 2884 genes over 17 conditions and is a widely used dataset to test the performance of blustering algorithms. Some values are missing in the yeast dataset, and are replaced by random values so that these random values could not form recognizable patterns and thus would be the leading candidates to get removed [1].The yeast dataset is available at http://arep.med.harvard.edu/biclustering/.

Biclustering is a NP-hard problem. With the number of genes and conditions increasing, the volume of the search space increases exponentially. When the number of genes increases, it gets harder to search the most optimal bicluster from the given data matrix.

To test the search efficiency of our proposed algorithm for greater data matrix, another yeast dataset consisting of 6601 genes and 17 conditions is also investigated.

B. Parameters

The main parameters in the proposed algorithm are set as follows:100 for the maximal number of iteration ,100 for the size of the population, 2.0 for the learning factor in Formula (1) and (3), 1 for the inertia weight in (1) and (3). The threshold δ used in the fitness function of (7) is set to 300 for

yeast dataset, it is a variational parameter with different dataset, so some experiments to optimize MSR are conducted in order to investigate the performance without the effect of the threshold δ .

In pattern-driven local search operator, a significant parameter α must be noted. The parameter has a strong impact on the size of the bicluster searched by the proposed algorithm. In the experiment, α is set to 0.8.

C. Results

1) Results for the original yeast dataset

In this paper, we concentrate on the search efficiency, this is to say the convergence, of the proposed algorithm. Fig.5 shows the convergence curves of the simple binary PSO algorithm and the proposed algorithm implemented on the yeast dataset with 2884 genes. Both of the figures in Fig.5 select the most fast convergent one among 10 biclusters found in experiments. In Fig.5 (a), the fitness rockets from a few tenths to nearly 5000 in the twenty-sixth generations, indicating that the MSR of the bicluster found by the simple binary PSO is reduced to the threshold δ at this condition point. In Fig.5 (b), the skyrocket of the convergence curve occur in the second generation, indicating that this approach can search the bicluster with MSR less than the threshold δ quickly.

In the process of searching, the local search operator is used to search the neighbors of each particle so that the search space is reduced fast. Consequently, the search efficiency has a great improvement. In order to show the gradient process, some experiments just to minimize the MSR are conducted, the results are shown in Fig.6.

2) Results for the extensional dataset

Another extensional yeast dataset with 6001 genes under 17 conditions is also investigated in experiments to show the search efficiency when the proposed algorithm is conducted on a more large data matrix. The extensional dataset has a much greater search space than the original yeast dataset has.

The convergence curves of simple binary PSO algorithm and the proposed algorithm for extensional dataset are shown in Fig.7, where x-axis represents the number of iterations while y-axis represents the fitness. In Fig.7, (a) shows two convergence curves of the simple binary PSO algorithm, it cannot reach 1 after 100 iterations, indicating that the MSR of bicluster found after 100 iterations is greater than δ, and (b) is two convergence curves of the proposed algorithm ,it can reach at least 90. Experimental results indicate that the pattern-driven local search can improve the search efficiency. Some detailed information about biclusters found by the proposed algorithm can obtained from Table I.

Fig.8 shows the profiles of several biclusters found by the proposed algorithm. These biclusters are chosen randomly from 10 biclusters obtained from experiments. In Fig.8, x-axis represents different conditions while y-axis represents expression values of genes. As can be seen from the figure, most of the genes in a bicluster have the similar trajectory to rise and fall at the same condition point.

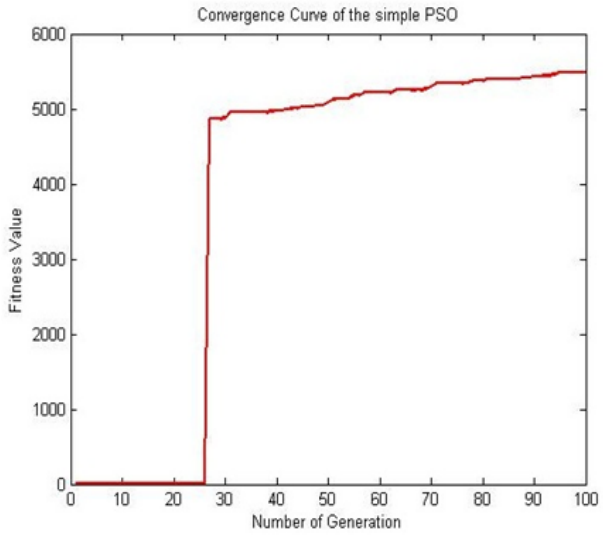
Table I gives the detailed information on the biclusters found by the proposed algorithm for the extensional yeast dataset when α is set to 0.8. As shown in Table I, the biclusters found by the proposed algorithm are satisfactory with respect to the MSR, but poor in size (volumes). Experimental results indicate that the size of the bicluster is controlled by the parameter α to a great extent. How the parameter α affects the size of a bicluster is in study now, so it is not discussed in this paper. This part will be a key point in our future work.

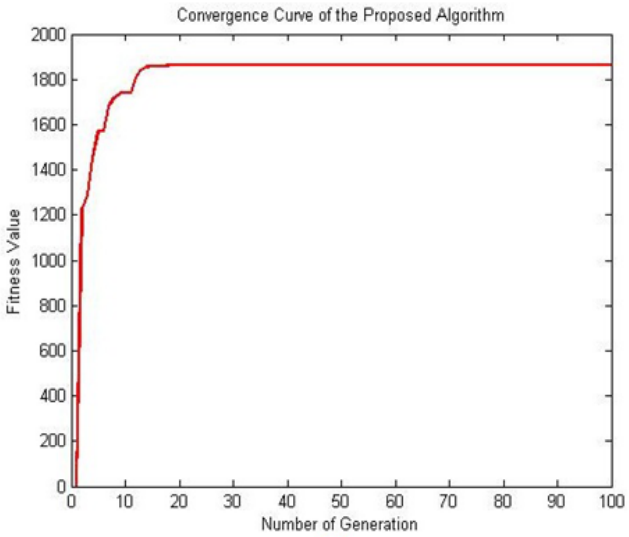
```
Input: Gene expression data matrix M, the maximal number of iteration maxIte, the number of bicluster to be found numBic, a threshold δ used in the fitness function computation, a threshold α used to constrain matching rate, the size of population popSize.
Output: The set of bicluster BiclusterSet.
Begin:
  num = 0;
  While num < numBic
    Initialize particle swarm position P of particle swarm with popSize particles;
    Initialize particle swarm velocity V with popSize particles;
    Initialize the history optimal position pbest and global optimal position gbest;
    numIte = 0;
    While numIte < maxIte
      Modify P to restrain the size of the bicluster determined by each particle.
      compute fitness;
      Update V according to (3);
      Update P according to (5);
      Do local search operator;
      Update V according to (3);
      Update P according to (5);
    End
    BiclusterSet = BiclusterSet∪gbest;
    num = num+1;
  End
End
```

Fig.3 Pseudo-code of the proposed algorithm. The parameter occurred in the algorithm are given in detail in Section IV.

```
Input: particles(biclusters)
Output: new particles after local searching
Begin:
Determine the bicluster pattern;
Removing bad genes;
Construct behavior matrix;
Search for neighbors;
Select genes from neighbors;
Return new bicluster.
end
```

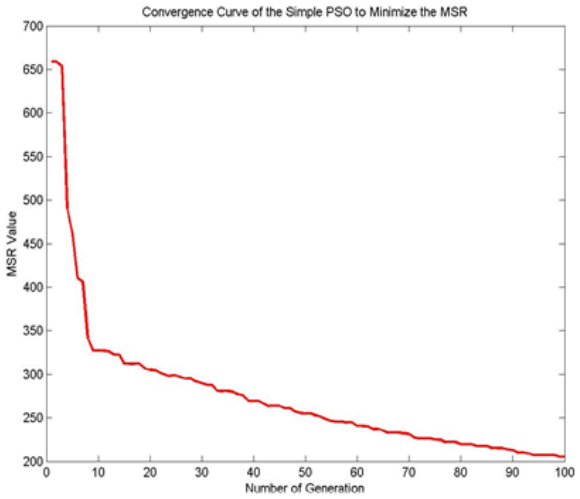
Fig.4 Pseudo-code of Local search operator.

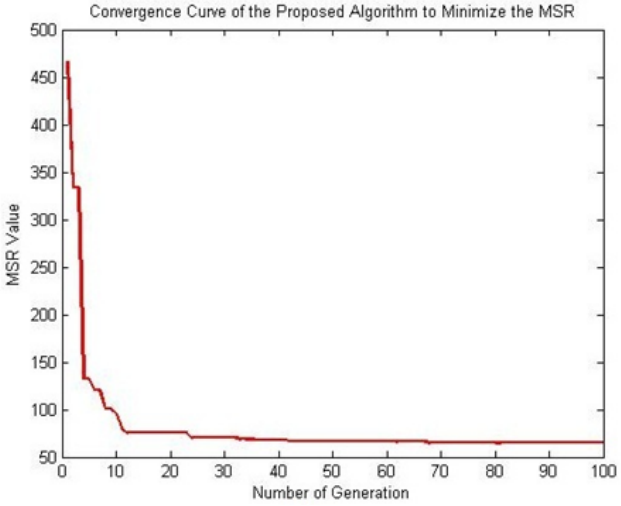




(a) (b)

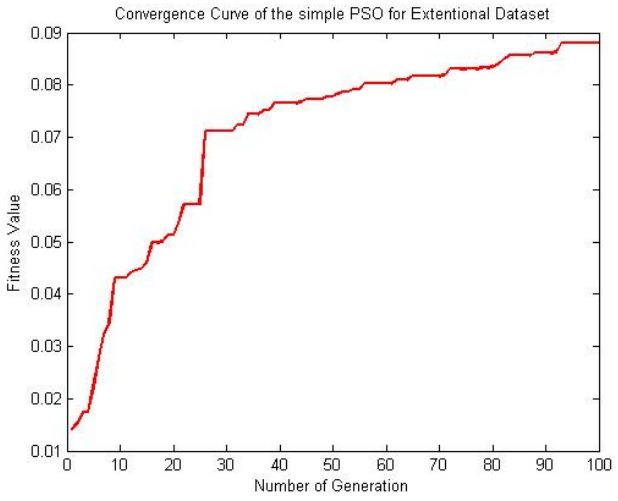
Fig.5 Convergence curves of the simple binary PSO and the proposed algorithm. (a) is the convergence of the simple binary PSO while (b) is the convergence of the proposed algorithm for the original yeast dataset. The convergence of the figures are the fast among 10 biclusters found by corresponding algorithms.

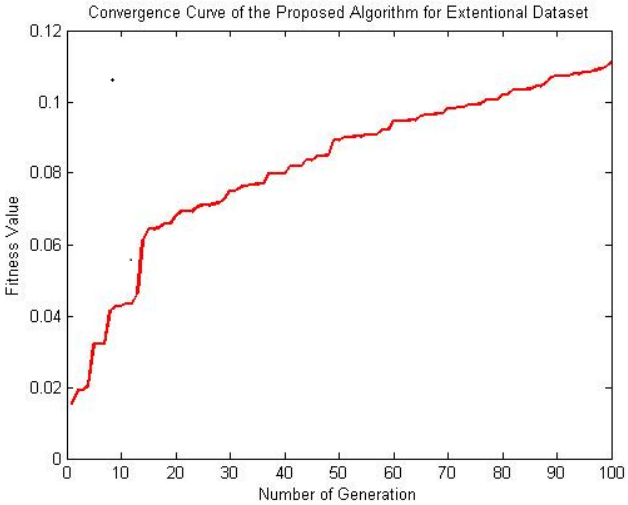




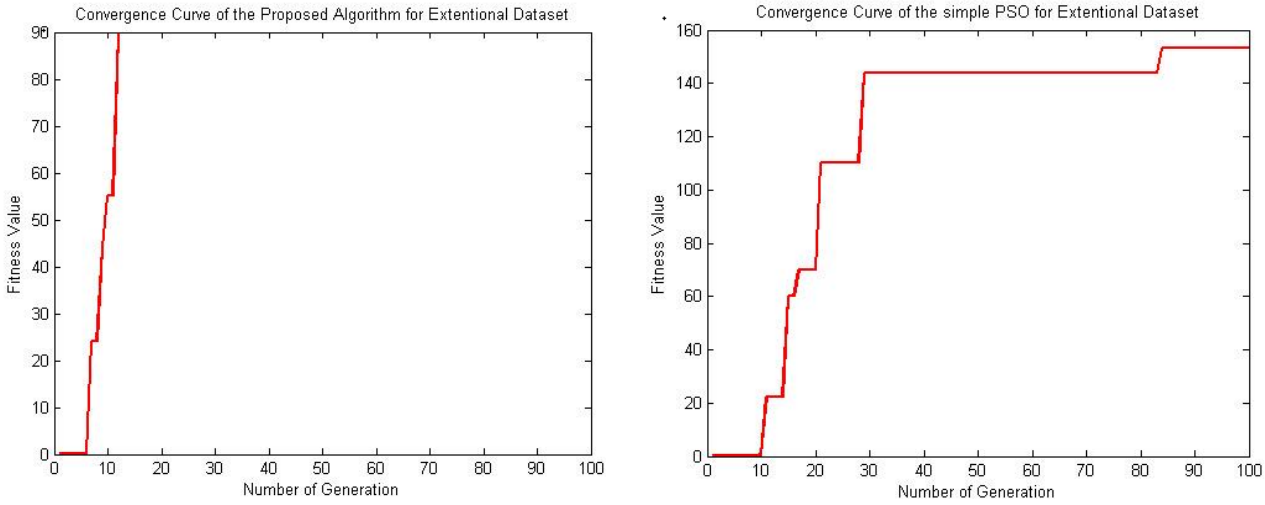
(a) (b)

Fig.6 Convergence curves of the simple binary PSO and the proposed algorithm. Both of the two figures are obtained from experiments to minimize the MSR for the original yeast dataset, (a) is the convergence of the simple binary PSO while (b) is the convergence of the proposed algorithm. The convergence of the figures are the best among 10 biclusters found by corresponding algorithms.



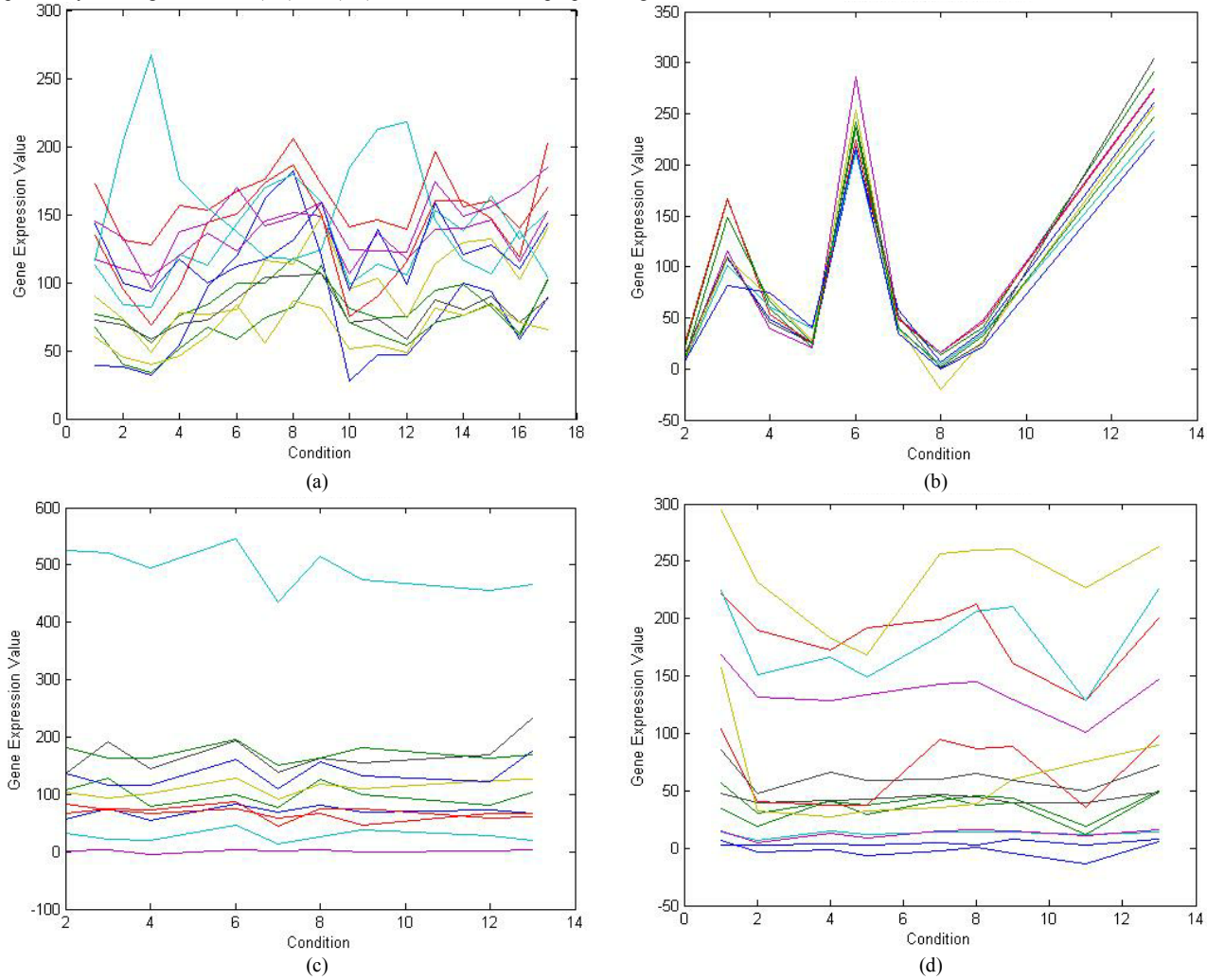


(a.1) (a.2)



(b.1) (b.2)

Fig.7 Convergence curves obtained by the simple binary PSO and the proposed algorithm from the extensional yeast dataset. (a.1) and (a.2) is obtained from the simple binary PSO algorithm, and (b.1) and (b.2) is obtained from the proposed algorithm.



(a) (b)

(c) (d)

Fig.8 Profiling of several biclusters obtained by the simple binary PSO and the proposed algorithm from the extensional yeast dataset. These biclusters are randomly chosen from biclusterset found.

TABLE I
INFORMATION ON BICLUSTERS

Id	number of genes	number of conditions	volumes	MSR score
Bi.1	13	17	221	**502**
Bi.2	9	9	81	208
Bi.3	463	3	1389	292
Bi.4	8	12	96	294
Bi.5	663	3	1989	**418**
Bi.6	9	10	90	291
Bi.7	11	9	99	221
Bi.8	13	9	117	230
Bi.9	14	9	126	279
Bi.10	10	9	90	279
mean value	121	9	1089	301

The volume is the product of the number of genes and the number of the conditions. Id is the index of the biclusters found by the proposed algorithm. Each value of the MSR reserves the integral part .These values in the above table are obtained when α is set to 0.8. The dark values are bad results, which should be reduced from result sets.

V. CONCLUSION

Biclustering is a significant technique for the analysis of gene expression data. In this paper, a novel biclustering algorithm based PSO integrated with a novel local search approach is proposed. The novel local search operator makes the most of the trajectory information of each gene over selected conditions to search from the neighborhood of a bicluster, and then an iterative adding operator is adopted in the local search operator. Experimental results indicate that the proposed algorithm can improve search efficiency.

Future work will be focused on the adjustment of the parameter α, the effect of this parameter on the size of the bicluster found, as well as the performance of the local search operator integrated into other intelligent optimization algorithm.

ACKNOWLEDGMENT

We thank Dr. Yang Wang for suggesting the use of the method of local search operator in intelligent optimization algorithms and assistance in the study. The authors would also like to thank the reviewers.

REFERENCES

[1] Y. Cheng, and G. M. Church, "Biclustering of expression data," In P*roceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, 93-103, 2000.

[2] A. Prelić, S. Bleuler, P. Zimmermann, etc., "A systematic comparison and evaluation of Biclustering methods for gene expression data," *Bioinformatics*, vol. 22, pp. 1122-1129, Feb. 2006.

[3] S. Busygin, O. Prokopyev, P. Pardalos, "Biclustering in data mining," *Computers and Operations Research* 2008, vol. 35, pp. 2964-2987, Sept. 2008.

[4] S. Bleuler, A. Prelic and E. Zitzler, "An EA framework for biclustering of gene expression data," *Congress on Evolutionary Computation CEC2004*, vol. 1, pp. 166-173, 19-23 June 2004.

[5] S. Jesus, Aguilar-Ruiz, and F. Divina, "Evolutionary Biclustering of micorarray data," *Lecture Notes in Computer Science*, vol. 3449, pp. 1-10, 2005.

[6] K. Bryan, P. Cunningham, N. Bolshakova, "Biclustering of Expression Data Using Simulated Annealing," *Proc. 18th IEEE Symposium. Computer-Based Medical System*s, pp. 383-388, June 2005.

[7] S. C. Madeira and A. L. Oliveira, "Biclustering algorithms for biological data analysis: a survey," *IEEE/ACM Transactions on Computational Biology and Bioinformatics,* vol. 1, pp. 24-45, Jan. - Mar. 2004.

[8] J. Kennedy, R. Eberhart, "Particle Swarm Optimization," P*roc. IEEE International Conf. on Neural Networks*, vol. 4, pp.1942 -1948, 1995 .

[9] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," *1997 IEEE Intl. Conf. on Systemics, Man, and Cybernetics* , vol. 5, pp. 4104-4108, Oct. 1997.

[10] L. Y. Chuang, H. W. Chang, C. J. Tu, C. H. Yang, "Improved binary PSO for feature selection using gene expression data," *Computational Biology and Chemistry*, vol. 32, pp.29-38, February 2008.

[11] W. Ayadi, M. Elloumi, J. K. Hao, "Pattern-driven neighborhood search for biclustering of microarray data," *Proceedings of the 2011 International Conference on Intelligent Computing (ICIC 2011)*, vol. 13, Aug. 2011.

[12] S. Mitra, H. Banka, "Multi-objective evolutionary biclustering of gene expression data," *Pattern Recognition*, vol. 39, pp. 2464-2477, Dec. 2006.

[13] A. Dharan, A. S. Nair, "Biclustering of gene expression data using reactive greedy randomized adaptive search procedure," *BMC Bioinformatics,* 2009, 10(Suppl. 1):S27.

[14] C. A. Brizuela, J. E. Luna-Taylor, I. Martinez-Perez, H. A. Guillen, "Improving an Evolutionary Multi-objective Algorithm for the Biclustering of Gene Expression Data," *2013 IEEE Congress on Evolutionary Computation*, pp. 221-228, June 2013.

[15] S. Mitra, H. Banka., "Multi-objective evolutionary biclustering of gene expression data," *ScienceDirect, Pattern Recognition*, vol. 39, pp. 2464 -2477, December 2006.

[16] J. A. Nepomuceno, A. Troncoso, J. S. Aguilar-Ruiz, "Biclustering of Gene Expression Data by Correlation-Based Scatter Search," *BioData Min.* Jan. 2011.

[17] J. S. Aguilar-Ruiz, "Shifting and scaling patterns from gene expression data," *Bioinformatics*, vol. 21, pp. 3840-3845, Aug. 2005.

[18] S. Das, S. M. Idicula, "Greedy Search-Binary PSO Hybrid for Biclustering Gene Expression Data," *International Journal of Computer Applications*, vol. 2, May 2010.