A Hybrid Metaheuristic for Biclustering Based on Scatter Search and Genetic Algorithms

Juan A. Nepomuceno[1], Alicia Troncoso[2], and Jesús S. Aguilar–Ruiz[2]

[1] Department of Computer Science, University of Sevilla, Spain
janepo@us.es
[2] Area of Computer Science, Pablo de Olavide University of Sevilla, Spain
{ali,aguilar}@upo.es

Abstract. In this paper a hybrid metaheuristic for biclustering based on Scatter Search and Genetic Algorithms is presented. A general scheme of Scatter Search has been used to obtain high–quality biclusters, but a way of generating the initial population and a method of combination based on Genetic Algorithms have been chosen. Experimental results from yeast cell cycle and human B-cell lymphoma are reported. Finally, the performance of the proposed hybrid algorithm is compared with a genetic algorithm recently published.

Keywords: Biclustering, Gene Expression Data, Scatter Search, Evolutionary Computation.

1 Introduction

Recently, data mining techniques are being applied to microarray data analysis in order to extract useful information [1]. Clustering techniques find groups of genes with similar patterns from a microarray. However, genes are not necessary related to every condition. Thus, the goal of the biclustering is to identify genes with the same behavior only under a specific group of conditions.

In the context of microarray analysis, many approaches have been proposed for biclustering [2]. Biclustering techniques have two important aspects: the search algorithm and the measure to evaluate the quality of biclusters.

Most of proposed approaches in the literature are focussed on different search methods. Thus, in [3] an iterative hierarchical clustering is applied to each dimension separately and biclusters are built by the combination of the obtained results for each dimension. In [4] an iterative search method which built biclusters adding or removing genes or conditions in order to improve the measure of quality called Mean Squared Residue (MSR) was presented. An exhaustive biclusters enumeration by means a bipartite graph-based model that nodes were added or removed in order to find maximum weight subgraphs was generated in [5]. The FLOC algorithm [6] improved the method presented in [4] obtaining a set of biclusters simultaneously and adding missing values techniques. In [7], a simple linear model for gene expression was used assuming normally distributed expression level for each gene or condition. Also, geometrical characterizations such as hyperplanes in a high dimensional data space have been used to find biclusters [8]. In the last few years, global optimization techniques, such as Simulated Annealing [9] or Evolutionary Computation [10,11], have been applied to obtain biclusters due to their good performance in several environments.

Recently, several papers were focussed on the measure proposed to evaluate the quality of biclusters. In [12] an analysis of the MSR was made, showing that this measure is good to find biclusters with shifting patterns but not scaling patterns. A new measure based on unconstrained optimization techniques was proposed in [13] as alternative to the MSR in order to find biclusters with certain patterns.

In this paper a hybrid metaheuristic for biclustering based on Scatter Search and Genetic Algorithms (SS&GA) is presented. A general scheme of Scatter Search has been used to obtain high–quality biclusters, but a way of generating the initial population and a method of combination based on Genetic Algorithms have been chosen. Finally, the performance of the proposed hybrid algorithm is compared with a genetic algorithm recently published [10]. A Scatter Search has been selected due to the recent success obtained to solve different hard optimization problems and to references about the application of Scatter Search for biclustering have not been found in the literature.

This paper is organized as follows. Section 2 presents basic concepts about Scatter Search. The description of the proposed metaheuristic is described in Section 3. Some experimental results from two real datasets and a comparison between the proposed method and a genetic algorithm are reported in Section 4. Finally, Section 5 outlines the main conclusions of the paper and future works.

2 Scatter Search

Scatter Search [14] is an optimization algorithm based on populations introduced in the seventies. Recently, Scatter Search algorithms have been applied to many nonlinear and combinatorial optimization problems providing remarkable outcomes due to its flexibility to adopt different search strategies mainly.

Basically, a standard Scatter Search can be summarized by the following steps:

1. Generate an initial population in a deterministic manner to assure the diversity of the population regarding a distance.
2. A set, called set of reference, is built with the best individuals from this population. The best individuals is not limited to a measure of individuals provided by a fitness function but a individual that improves the diversity can be added to the reference set.
3. New individuals are created by the deterministic combination of individuals of the reference set and all individuals of the reference set are selected to be combined.
4. The reference set is updated using the new individuals and the combination is repeated until the reference set does not change.
5. The reference set is rebuilt and if the maximum number of iterations is not reached go to step 3.

Therefore, the search strategies of a Scatter Search depend on a diversification method to generate the initial population, a method to built the reference set, a method to combine individuals and a method to rebuilt the reference set.

The main differences between a Genetic Algorithm and a Scatter Search are the way of generating the initial population, as the initial population is generated randomly and deterministic, respectively; the selection of individual to create offspring, as a probabilistic procedure is applied to select parents in Genetic Algorithms and all individuals of the reference set are combined in Scatter Search; the evolution of the population, based on the survival of the best depending on the fitness function in Genetic Algorithms and the rebuilding method of reference set used in Scatter Search. Finally, the size of the population in Genetic Algorithms is bigger than that of the reference set in Scatter Search. A typical size in Genetic Algorithms is 100 and 10 in Scatter Search, due to that the combination method in Scatter Search takes into account all pairs of individuals to create new individuals. In short, the underlying idea of Scatter Search is to emphasize systematic processes against random procedures to generate populations, to create new individuals and to inject diversity to the population.

3 Description of the Algorithm

In this section the proposed SS&GA algorithm to obtain biclusters is described, detailing the steps aforementioned in the previous section such as combination, generation, updating and rebuilding methods.

The pseudocode of the proposed SS&GA algorithm is presented in algorithm 1.

3.1 Biclusters Codification and Generation

Formally, a microarray is a real matrix composed by N genes and M conditions. The element (i, j) of the matrix means the level of expression of gene i under the condition j. A bicluster is a submatrix of the matrix M composed by $n \leq N$ rows or genes and $m \leq M$ columns or conditions.

Biclusters are encoded by binary strings of length $N + M$ [10]. Each of the first N bits of the binary string is related to the genes and the remaining M bits to the conditions from microarray M. For instance, the bicluster shown in Fig. 1 is encoding by the following string,

$$0010110000|01100 \tag{1}$$

Thus, this string codifies the bicluster composed by genes number 3, 5 and 6 and conditions 2 and 3 from a microarray comprising 10 genes and 5 conditions.

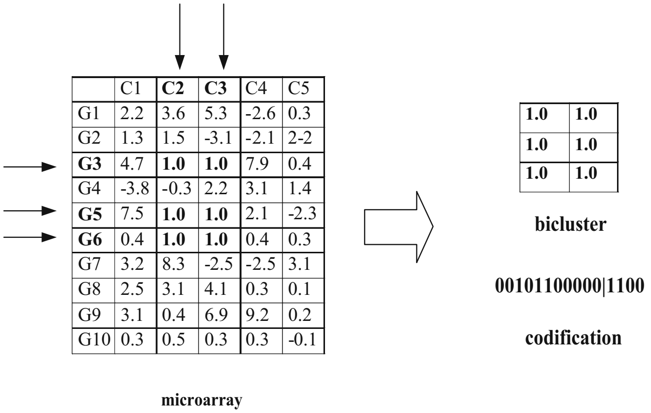
The initial population of biclusters is strictly randomly generated (typical in Genetic algorithms) without taking into account the diversity (typical in Scatter Search). Random strings composed by 0 and 1 are generated until nB biclusters are built, where nB is the size of the starting population, i.e. the number of biclusters.

Algorithm 1. *SS&GA* FOR BICLUSTERING

```
INPUT Microarray $M$, penalization factors $M_1$ and $M_2$, size of population $nB$, size
   of reference set, $S$, and maximum number of iterations, $MaxIter$.
OUTPUT The reference set, $RefSet$
begin
  Initialize $P$ randomly with $nB$ biclusters
  //Building reference set
  $R_1 \leftarrow S/2$ best biclusters from $P$ (according to their fitness function)
  $R_2 \leftarrow S/2$ most scatter biclusters, regarding $R_1$, from $P \setminus R_1$ (according to a distance).
  $RefSet = R_1 \cup R_2$
  $P = P \setminus RefSet$
  //Initialization
  stable $\leftarrow$ FALSE
  $iter = 0$
  while ($iter \leq MaxIter$) do
    //Updating reference set
    while (NOT stable) do
      $A \leftarrow RefSet$
      $B \leftarrow CombinationMethod(RefSet)$
      $RefSet \leftarrow S$ best biclusters from $RefSet \cup B$
      if ($A = RefSet$) then
        $stable \leftarrow TRUE$
      end if
    end while
    //Rebuilding reference set
    $R_1 \leftarrow S/2$ best biclusters from $RefSet$ (according to their fitness function)
    $R_2 \leftarrow S/2$ most scatter biclusters, regarding $R_1$, from $P \setminus R_1$.
    $RefSet = R_1 \cup R_2$
    $P \leftarrow P \setminus RefSet$
    $iter = iter + 1$
  end while
end
```

3.2 Building Reference Set

The reference set comprises the best S biclusters of the initial population, P, where S is the number of biclusters that belong to this set. The reference set is built taken into account both quality and scattering of biclusters. The quality of biclusters is measured evaluating the fitness function considered in the evolutive process. Thus, a bicluster is better than another if the fitness function value is lower than that of the second one. On the other hand, a distance must be defined in order to show how the scattering is introduced in the search space. In the proposed SS&GA approach the distance used is the *Hamming* distance. The *Hamming* distance between two binary strings is defined by the number of positions for which their corresponding 0/1 values are different. For example, the *Hamming* distance for strings 001001001|001 and 001011001|101 is 2.



	C1	**C2**	**C3**	C4	C5
G1	2.2	3.6	5.3	-2.6	0.3
G2	1.3	1.5	-3.1	-2.1	2-2
G3	4.7	**1.0**	**1.0**	7.9	0.4
G4	-3.8	-0.3	2.2	3.1	1.4
G5	7.5	**1.0**	**1.0**	2.1	-2.3
G6	0.4	**1.0**	**1.0**	0.4	0.3
G7	3.2	8.3	-2.5	-2.5	3.1
G8	2.5	3.1	4.1	0.3	0.1
G9	3.1	0.4	6.9	9.2	0.2
G10	0.3	0.5	0.3	0.3	-0.1

1.0	**1.0**
1.0	**1.0**
1.0	**1.0**

Fig. 1. Microarray and bicluster along with its codification

Therefore, the reference set is formed by the $S/2$ best biclusters from P (set R_1) according to their fitness function and the $S/2$ biclusters from $P \setminus R_1$ (set R_2) with the highest distances to the set R_1 according to the *Hamming* distance.

3.3 Combination Method and Updating Reference Set

Combination method is the mechanism to create new biclusters in Scatter Search. All pairs of biclusters are combined generating $S*(S-1)/2$ new biclusters. In the SS&GA algorithm the typical uniform crossover operator used in Genetic Algorithms is the proposed combination method. This crossover operator is shown in Fig. 2. A binary mask is randomly generated and a child is composed by values from the first parent when there is a 1 in the mask, and from the second parent when there is a 0.

The reference set is updated with the S best biclusters from the reference set and the new biclusters generated by the combination method according to the fitness function. This process is repeated iteratively until the reference set does not change.

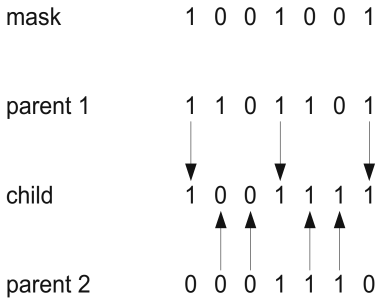


Fig. 2. Uniform crossover operator of Genetic Algorithms

3.4 Rebuilding Reference Set

After getting the stability of reference set in the updating process, this set is rebuilt to introduce diversity in the search process. This task is made by mutation operators in Genetic Algorithms. Thus, the reference set is composed by the $S/2$ best biclusters from the updated reference set (set R_1) according to the fitness function and the $S/2$ most distant from $P \setminus R_1$ according to the *Hamming* distance.

3.5 Biclusters Evaluation

The fitness function is fundamental in order to evaluate the quality of biclusters. Cheng and Church proposed the MSR which measures the correlation of a bicluster. Given a bicluster comprising the subset of genes I and the subset of conditions J, the MSR is defined as follows,

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

where

$$R(i,j) = e_{ij} - e_{Ij} - e_{iJ} + e_{IJ} \tag{3}$$

$$e_{Ij} = \frac{1}{|I|} \sum_{i \in I} e_{ij} \tag{4}$$

$$e_{iJ} = \frac{1}{|J|} \sum_{j \in J} e_{ij} \tag{5}$$

$$e_{IJ} = \frac{1}{|IJ|} \sum_{i \in I, j \in J} e_{ij} \tag{6}$$

In this work, biclusters with low residue and high volume are preferred. Therefore, the fitness function is defined by:

$$f(B) = MSR(B) + M_1 \left(\frac{1}{G}\right) + M_2 \left(\frac{1}{C}\right) \tag{7}$$

where $MSR(B)$ is the MSR of the bicluster B, M_1 and M_2 are penalization factors to control the volume of the bicluster B, and G and C are the number of genes and conditions of the bicluster B, respectively.

The use of MSR in the fitness function considered in the proposed SS&GA algorithm allows to establish a comparison with a previous evolutionary-based biclustering method and the Cheng and Church algorithm.

4 Experimental Results

Two well known datasets [4] have been used to shows the performance of the proposed SS&GA algorithm. The first dataset is the *yeast Saccharomyces cerevisiae* cell cycle expression and the second one is the *human B-cells* expression data originated from [15] and [16], respectively. Original data were preprocessed in [4] replacing missing values with random numbers. The Yeast dataset contains 2884 genes and 17 experimental conditions and the Human dataset consists of 4026 genes and 96 conditions.

The main parameters of the proposed SS&GA algorithm are as follows: 200 for the initial population; 10 for the reference set and 20 for the maximum number of iterations. The penalization factor for the number of conditions has been chosen of one order of magnitude larger than the range in which the fitness function varies for both datasets. However, that of the number of genes has been chosen of same order of magnitude to the range of values of the fitness function for both datasets. The main goal of this choice is to test the influence of the penalization factors on the volume of the biclusters.

4.1 Yeast Data Set

Table 1 presents several biclusters obtained by the application of the SS& GA approach from Yeast dataset. For each bicluster is shown an identifier of the bicluster, the value of its MSR, the number of genes and the number of conditions. It can be observed that high–quality biclusters have been obtained as the values of the MSR are lower than 220. Moreover, the volume of the obtained biclusters is satisfactory showing that the SS& GA approach find non–trivial biclusters. Concretely, biclusters and no clusters are obtained since the number of conditions is less than 17 always.

Biclusters presented in Table 1 are shown in Fig. 3. Although biclusters are good taking into account the values of their MSR, in this figure their tends cannot be observed easily. This is due to the overlapping among biclusters as the same gene can be found in different biclusters.

Fig. 4 shows the evolution of the average MSR, fitness function values and volume of the reference set throughout the evolutionary process for the Yeast dataset. The values of the MSR and the volume are represented in the axis on the left and that of the fitness function in the axis on the right. It can be noticed that the initial reference set improves the average MSR throughout the iterations

Table 1. Results obtained by SS&GA algorithm for Yeast dataset

Bicluster	MSR	Genes	Conditions
bi.1	74.72	10	13
bi.2	106.25	13	13
bi.3	125.9	22	13
bi.4	216.16	25	14
bi.5	97.04	26	11
bi.6	117.25	14	14
bi.7	136.67	25	13
bi.8	159.44	39	13
bi.9	121.89	26	11

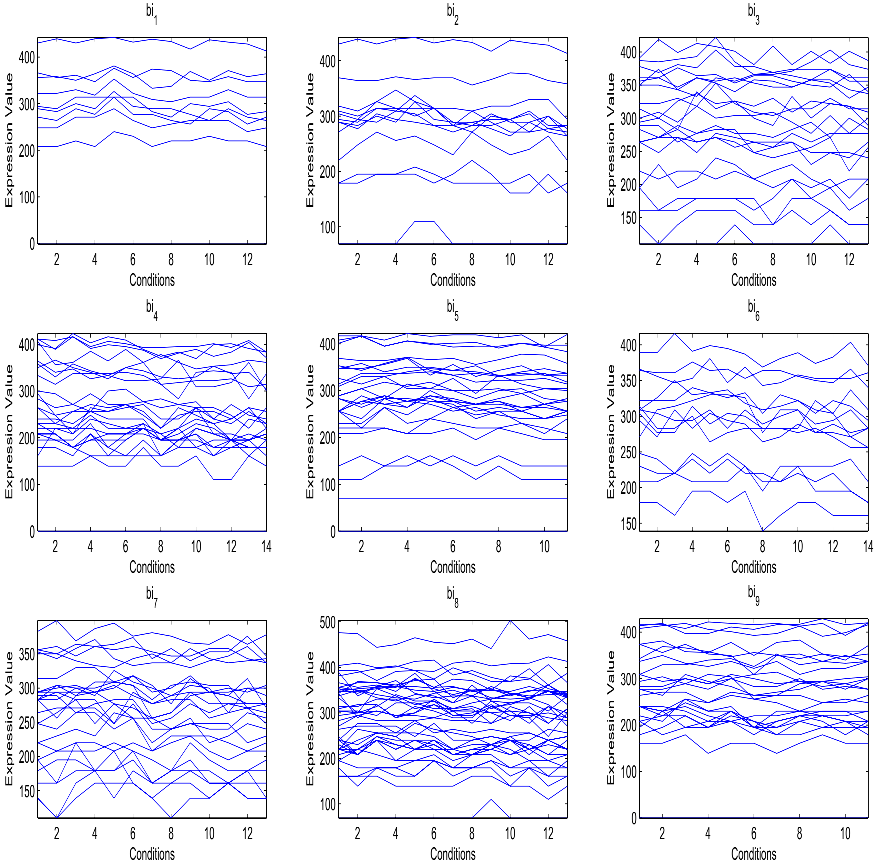


Fig. 3. Biclusters from Yeast dataset

and the SS&GA algorithm converges in 8 iterations approximately. The average volume of the reference set decreases versus the number of iterations due to the non too large penalization factors have been chosen.

4.2 Lymphoma Data Set

Table 2 presents information about several biclusters found by the SS&GA approach for Human dataset. The values of the MSR are considerably low since all are lower than 1100. Thus, it can be stated that obtained biclusters have a remarkable quality. Moreover, in general the obtained biclusters have a large number of genes, specially the bicluster number 1, 2 and 4. These biclusters are also represented in Fig. 5.

Figure 6 presents the performance of the proposed algorithm for Human dataset. The evolution of the average MSR, fitness function values and volume

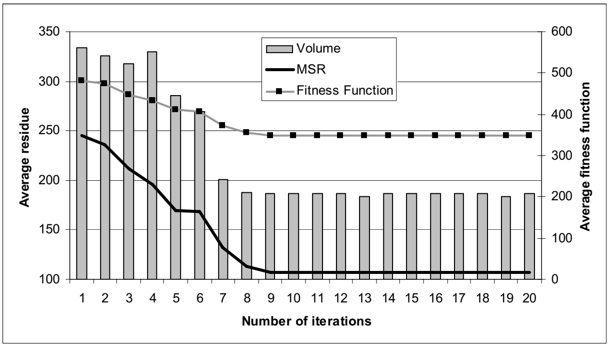


Fig. 4. Performance of the proposed SS&GA algorithm for Yeast dataset

Table 2. Results obtained by SS&GA for Human dataset

Bicluster	MSR	Genes	Conditions
bi.1	855.17	109	13
bi.2	813.70	127	12
bi.3	642.13	85	11
bi.4	815.74	122	10
bi.5	771.69	48	12
bi.6	595.69	44	9
bi.7	1074.10	56	13
bi.8	507.17	67	8
bi.9	794.07	70	11

Table 3. Comparison of the results obtained by SS&GA, SEBI and CC algorithms

Algorithm-Dataset	Avg. Residue	Avg. gene num.	Avg. cond. num.
SS&GA–Yeast	128.37 (40.71)	22.23 (8.86)	12.78 (1.09)
SS&GA–Human	763.27 (165.73)	80.89 (36.61)	11 (1.73)
SEBI–Yeast	205.18 (4.49)	13.61 (10.38)	15.25 (1.37)
SEBI–Human	1028.84 (29.19)	14.07 (5.39)	43.57 (6.20)
CC–Yeast	204.29 (42.78)	166.71 (226.37)	12.09 (4.39)
CC–Human	850.04 (153.91)	269.22 (204.71)	24.5 (20.92)

for the reference set is shown. A good performance of the SS&GA technique and a fast convergence can be appreciated. The values of the fitness function decrease quickly and only ten iterations approximately are needed to find high–quality biclusters. In this case, the choice of penalization parameters to keep under control the volume of the biclusters provides a nearly constant volume in the last iterations.

Finally, a comparison between the results obtained with the SS&GA algorithm and two representative techniques reported in the literature is provided. Concretely, the SS&GA algorithm is compared to SEBI [10] and Cheng and

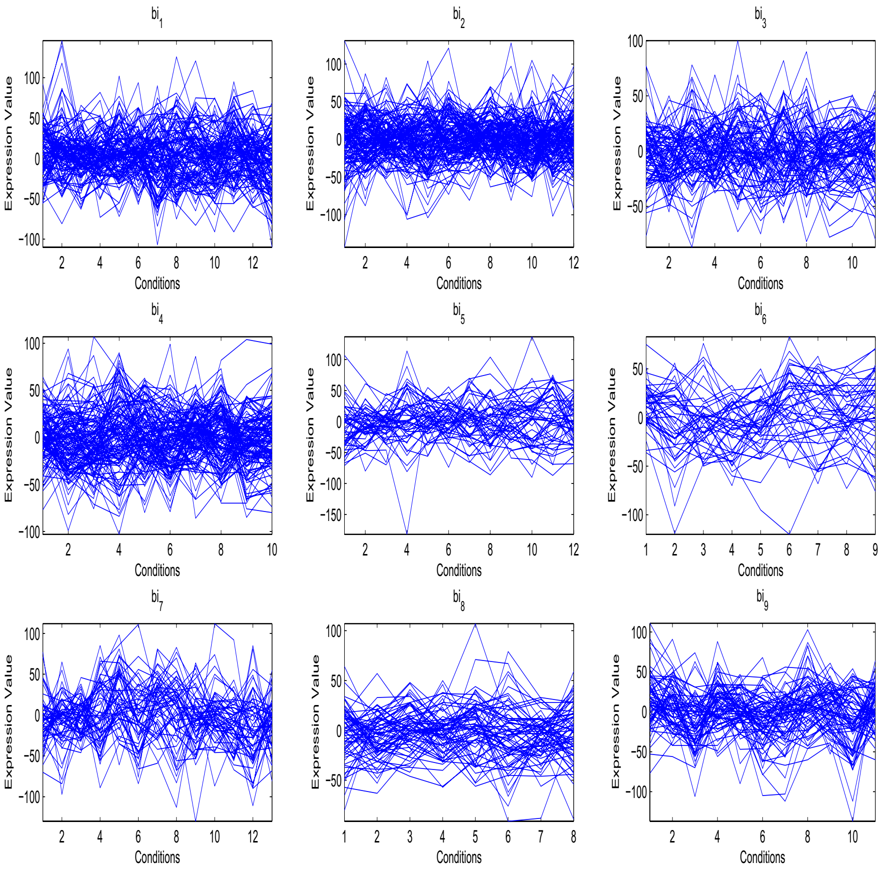


Fig. 5. Biclusters from Human dataset

Church (CC) algorithm [4]. The SEBI approach is a genetic algorithm which introduces mechanisms to avoid the overlapping among biclusters. On the other hand, the most of biclusters obtained by the CC algorithm are overlapped.

Table 3 presents the average of the MSR and the average of the number of genes and conditions of the biclusters found by the three approaches. Furthermore, the standard deviation is shown in brackets. It can be observed that the proposed algorithm improves all the average MSR in spite of SEBI obtains biclusters smaller than CC and SS&GA methods. The small volume of the biclusters found by SEBI algorithm, due to the control of the overlapping, should provide a lower MSR. As regards the standard deviation, it is the SEBI approach which has a more stable behavior since CC and SS&GA methods have standard deviations larger than the SEBI algorithm. In short, it can be stated that the SS&GA algorithm has a good performance yielding competitive results with respect to that of other techniques.

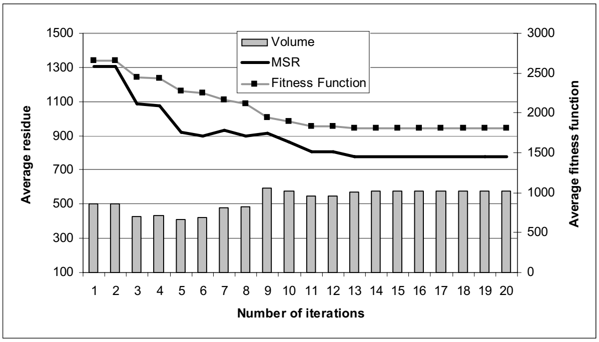


Fig. 6. Performance of the proposed SS&GA algorithm for Human dataset

5 Conclusions

A hybrid metaheuristic for biclustering based on Scatter Search and Genetic Algorithms has been presented in this work. A general scheme of Scatter Search has been used to obtain high–quality biclusters, but the starting population has been generated randomly and an uniform crossover operator to create new biclusters has been chosen from Genetic Algorithms. Experimental results from yeast cell cycle and human B-cell lymphoma have been reported and the outcomes of the proposed hybrid algorithm has been compared with that of a genetic algorithm, showing a satisfactory performance taking into account the difficulty of the biclustering problem.

Future works will be focussed on the use of deterministic combination methods and diversification methods to generate the initial population. Moreover, other measures based on scaling and shifting patterns to evaluate biclusters will be tested.

Acknowledgments

The financial support given by the Spanish Ministry of Science and Technology, project TIN-68084-C02-01 and by the Junta de Andalucía, project P07-TIC-02611 is acknowledged.

References

1. Larranaga, P., et al.: Machine learning in bioinformatics. Briefings in Bioinformatics 7(1), 86–112 (2006)
2. Busygin, S., Prokopyev, O., Pardalos, P.M.: Biclustering in data mining. Computers and Operations Research 35(9), 2964–2987 (2008)
3. Levine, E., Getz, G., Domany, E.: Couple two-way clustering analysis of gene microarray data. Proceedings of the National Academy of Sciences (PNAS) of the USA 97(22), 12079–12084 (2000)

4. Cheng, Y., Church, G.M.: Biclustering of Expression Data. In: 8th International Conference on Intelligent Systems for Molecular Biology, pp. 93–103 (2000)
5. Tanay, A., Sharan, R., Shamir, R.: Discovering statistically significant biclusters in gene expression data. Bioinformatics 18(1), 136–144 (2002)
6. Yang, J., Wang, H., Wang, W., Yu, P.: Enhanced biclustering on expression data. In: 3rd IEEE Symposium on Bioinformatics and Bioengineering, pp. 321–327 (2003)
7. Bergmann, S., Ihmels, J., Barkai, N.: Iterative signature algorithm for the analysis of large-scale gene expression data. Physical Review E 67(3), 31902 (2003)
8. Harpaz, R., Haralick, R.: Exploiting the geometry of gene expression patterns for unsupervised learning. In: 18th International Conference on Pattern Recognition (ICPR 2006), pp. 670–674 (2006)
9. Bryan, K., Cunningham, P., Bolshakova, N., Coll, T., Dublin, I.: Biclustering of expression data using simulated annealing. In: 18th IEEE International Symposium on Computer-Based Medical Systems, pp. 383–388 (2005)
10. Divina, F., Aguilar-Ruiz, J.S.: Biclustering of Expression Data with Evolutionary Computation. IEEE Transactions on Knowledge and Data Engineering 18(5), 590–602 (2006)
11. Mitra, S., Banka, H.: Multi-objective evolutionary biclustering of gene expression data. Pattern Recognition 39(12), 2464–2477 (2006)
12. Aguilar-Ruiz, J.S.: Shifting and scaling patterns from gene expression data. Bioinformatics 21(20), 3840–3845 (2005)
13. Nepomuceno, J.A., Troncoso, A., Aguilar-Ruiz, J.S., Garcıa-Gutierrez, J.: Biclusters Evaluation Based on Shifting and Scaling Patterns. In: Yin, H., Tino, P., Corchado, E., Byrne, W., Yao, X. (eds.) IDEAL 2007. LNCS, vol. 4881, pp. 840–849. Springer, Heidelberg (2007)
14. Marti, R., Laguna, M.: Scatter Search. Methodology and Implementation in C. Kluwer Academic Publishers, Boston (2003)
15. Cho, R.J., et al.: A Genome-Wide Transcriptional Analysis of the Mitotic Cell Cycle. Molecular Cell 2(1), 65–73 (1998)
16. Alizadeh, A.A., et al.: Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. Nature 403, 503–511 (2000)