

Extracting Additive and Multiplicative Coherent Biclusters with Swarm Intelligence

Fabício O. de França*, Fernando J. Von Zuben†

Laboratory of Bioinformatics and Bioinspired Computing (LBiC), Department of Computer Engineering and Industrial Automation (DCA), School of Electrical and Computer Engineering (FEEC), University of Campinas (Unicamp)
Av. Albert Einstein, 400 – SP - 13083-852, Brazil
Email: *olivetti@ieee.org, †vonzuben@dca.fee.unicamp.br

Abstract—Biclustering is usually referred to as the process of finding subsets of rows and columns from a given dataset expressing a relationship. Each subset is a bicluster and corresponds to a sub-matrix whose elements tend to present a high degree of coherence with each other, that may lead to novel discoveries regarding the objects in the dataset. This coherence leads to the possibility of obtaining representative values for rows (subset of objects) and columns (subset of attributes) of each bicluster. In the literature, it is usually studied the additive coherence among elements, i.e. each element is represented by the sum of its respective representative values. But in a given dataset, it is also possible to find multiplicative relations, i.e. each element being represented by the multiplication of its respective representative values, and that may reveal distinct knowledge contained in the objects of the dataset. So, in this paper, a swarm-based approach, named *SwarmBcluster*, is adapted to find both additive and multiplicative coherent biclusters from a dataset, in an attempt to enrich the amount of information provided by the biclusters. Experiments are performed considering two well-known datasets and it is found that the multiplicative coherence biclusters improve the quality of the data analysis and may contribute to reduce the influence of noise.

I. INTRODUCTION

The ever-increasing capability of acquiring data from different sources, developed by different fields of knowledge, has led to a growth in the amount of available data. In order to help data analysts and researchers to extract knowledge from this high amount of information the data mining field has evolved, in the past years, to produce advanced tools to treat these data and the available information more efficiently.

Within data mining, a very important field is *clustering*, which consists of identifying objects that can be grouped together, according to some similarities in their attributes. Several classical tools to perform clustering, such as *k*-means, Self Organized Maps (SOMs) and Hierarchical Clustering have been proposed in the literature, and successfully applied to different kinds of problems [1]. However, these classical techniques also present some limitations when dealing with large and heterogeneous datasets: they cannot detect partial matching since they group data based on global similarities; and most of these methods are only capable of assigning a given object to only a single category (cluster). The capability of identifying that two objects present similarities when only

a subset of their attributes is considered can be useful to perform deeper inferences from the data, while the assignment of a given object to only one category may be insufficient for several problems, ranging from text mining to ecological networks in biology [2].

With the problem faced by those algorithms, a new technique that performs the clustering process simultaneously over the objects and attributes of the data matrix was proposed in the literature. This procedure, which was called *biclustering* [3], is capable of finding subsets of objects which are somehow correlated to subsets of attributes, allowing the extraction of additional information from the dataset.

Given that distinct biclusters express independent relationships in a dataset, it is desirable to produce multiple biclusters for a better analysis of the whole data. The problem of finding a set of biclusters may be considered similar to the problem of finding several two-way bipartitions of the whole dataset, which is clearly a combinatorial optimization problem. Also, the construction of a bicluster leads to a compromise between two conflicting objectives: it is required that the volume, often calculated as the product between the number of rows and the number of columns of the bicluster, is maximized, and that the similarity among its elements is kept as high as possible.

The concept of δ -bicluster, introduced in [3], denotes the problem of finding a set of biclusters that have a degree of coherency, measured by a mean-squared residue error, no smaller than a given threshold while maximizing the volume of such biclusters. The degree of coherency was the additive coherence measure, although it was pointed out in [3] that it would also be possible to find multiplicative coherence by simply transforming the data by calculating the logarithm of each element.

Some related work have successfully applied a multiplicative coherence measure, but, so far, most of these algorithms are able to concentrate the search only on one type of coherency at once. So, in this paper, a swarm-based approach, called *SwarmBcluster* [4], that was created to find a bicluster set with additive coherency, is adapted to be able to find additive and multiplicative coherent biclusters concurrently. It will also be shown that, differently from other approaches, this algorithm is capable of deciding what kind of coherency is more appropriate for each region of the dataset.

The authors would like to thank CAPES and CNPq for the financial support.

This paper is organized as follows. Section II introduces the main concepts of biclustering. The coherence measure is introduced in Section III together with an explanation on how to transform a multiplicative coherent bicluster into an additive one. The swarm-based approach, SwarmBcluster, will be briefly explained in Section IV together with the adaptations necessary to deal with the multiplicative coherent measure and other measures as well. The experiments performed and the comparison among SwarmBcluster with and without multiplicative coherency will be shown in Section V. Finally, the concluding remarks of this paper are presented in Section VI.

II. BICLUSTER

In data mining, biclustering is referred to the process of finding subsets of rows and columns of a given data matrix with elements expressing correlation [3]. This data matrix may represent different kinds of numerical data, such as objects and its attributes, respectively comprising the rows and columns of the matrix.

There is a wide scope of different applications that require an algorithm capable of finding a set of biclusters. The main motivation is to find data points that are correlated under only a subset of the attributes, which is above the capabilities of usual clustering methods. Some examples of biclustering applications are dimensionality reduction [5], information retrieval and text mining [6], [7], electoral data analysis [8], collaborative filtering [9] and biological data analysis [5]. In this paper, the biclustering approach will be employed to find coherence inside biological data on microarray experiments.

Several approaches were proposed in the literature to extract biclusters from a dataset [2], [6], [5], [3], [10], [11]. Since the term *biclustering* refers loosely to a group of data analysis approaches, and it is not a defined problem *per se*, those approaches differ on how they formulate the biclustering search problem, and thus on how they find different types of biclusters.

One aspect that influences the problem formulation is how the elements of a bicluster are correlated. For instance, some algorithms search for constant valued biclusters (like the ones in Fig. 1(a)), some for constant columns or rows, and others for coherency in the values of the elements. In Fig. 1, biclusters expressing precise instances of usually adopted quality measures are illustrated. In practice, however, there will generally be a deviation from the desired profile, called the residue of the bicluster.

Another aspect of a bicluster that must be taken into consideration is its *volume*, as it is commonly called, and it is calculated by multiplying the number of rows by the number of columns of such bicluster. The larger the volume of a bicluster, the more information it contains, specially if there exists a high coherence among its elements. It should be noticed by now that the quality measure of a bicluster and its volume lead to a multi-objective problem in which one would want to find a bicluster with high volume and better quality but, as the bicluster volume becomes higher the quality will most likely become worse, thus characterizing two conflicting objectives.

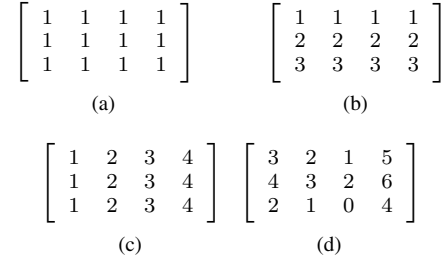


Fig. 1. Example of different quality measures: (a) constant bicluster, (b) constant rows, (c) constant columns, (d) coherent values (rows or columns should exhibit high correlation).

The multi-objective formulation was well addressed in [11] with a corresponding algorithm to solve it.

Since most approaches usually find a set of such biclusters, two other aspects must be taken into consideration as well. One of them is the maximum overlap between two biclusters, i.e., the amount of elements common to two biclusters from the set, thus avoiding redundancy of information. The other aspect is the coverage of the dataset, or how much elements from the dataset belongs to at least one bicluster from the set. By maximizing the coverage of the set, it is possible to achieve a more diverse information regarding different portions of the dataset.

The next section will focus on the metric adopted in the bicluster searching problem, denoted coherence measure, that will be the object of study in this paper.

III. COHERENCE MEASURE

In biclustering applications, coherence means that the elements inside a given bicluster, with size $n \times m$, have a predictable behavior. As a consequence every element of a bicluster may be calculated by using a representative value for the bicluster's rows ($\alpha_i, i \in \{1, \dots, n\}$) and for the bicluster's columns ($\beta_j, j \in \{1, \dots, m\}$), taking into consideration a constant bias (μ) specific to this bicluster. A common formula to calculate such elements is the additive coherence, which express an additive relation between rows or columns, such as the one illustrated on Fig. 1(d):

$$a_{i,j} = \alpha_i + \beta_j - \mu, \quad (1)$$

where $a_{i,j}$ is the element (i, j) of the bicluster.

Another possible behavior that a bicluster might exhibit is the multiplicative coherence, in which each row or column is a factor of the elements of other rows and columns, expressed by:

$$a_{i,j} = \alpha_i \times \beta_j \div \mu. \quad (2)$$

It is worth noticing though that, by transforming the representative values of a multiplicative bicluster by calculating their logarithm, the multiplicative coherence changes to an additive coherence, and the real outcome turns to be the logarithm of the calculated element. In Fig. 2 it is plotted

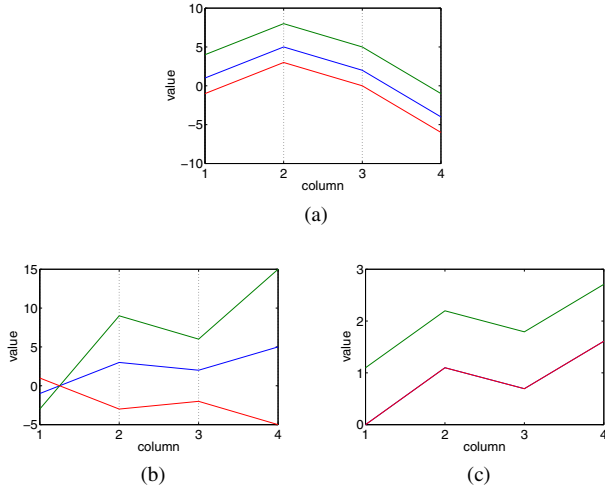


Fig. 2. Elements expressing similar behavior with: (a) additive coherence, (b) multiplicative coherence, and (c) logarithm of the multiplicative coherence.

the behavior of three rows of an additive, multiplicative and the logarithm of the multiplicative coherent bicluster. Notice that, since one of the rows from the multiplicative coherent bicluster was the negative of another row, the logarithm was calculated from the absolute value, thus making them the same line on Fig. 2c (the lower line).

In [3], Cheng and Church had devised a way of finding the representative values of a given bicluster. By assuming that each row or column of the bicluster presents a profile identical to the one exhibited by other rows or columns, except for a constant bias, each element $a_{i,j}$ of this matrix can be calculated by:

$$a_{i,j} = a_{I,j} + a_{i,J} - a_{I,J}, \quad (3)$$

where $a_{I,j}$ is the mean value of the j -th column of the bicluster, $a_{i,J}$ is the mean value of the i -th row, and $a_{I,J}$ is the mean value considering all the elements of the bicluster.

So, the problem of finding an additive coherent bicluster can be formulated as the problem of finding a bicluster that minimizes the error between the calculated value and the real value of an element of the matrix. The mean squared residue (MSR), proposed in [3], is formulated as:

$$H(I, J) = \frac{1}{|I||J|} \sum_{i,j \in I, J} r_{i,j}^2, \quad (4)$$

where $|I|$ is the total number of rows of the bicluster, $|J|$ is the total number of columns of the bicluster, and $r_{i,j}$ is given by

$$r_{i,j} = a_{i,j} - a_{I,j} - a_{i,J} + a_{I,J}. \quad (5)$$

By introducing this quality measure, Cheng and Church also introduced the concept of δ -bicluster. The corresponding problem consists of finding, inside a dataset, a maximal volume bicluster with the MSR constrained to a value lower than or equal to a given δ .

A. Multiplicative Coherence

As mentioned before, in order to obtain the multiplicative coherence for the elements of a bicluster, it is possible to calculate the logarithm of the reference values, thus obtaining the logarithm of a given element by using Eq. 3. However, this data transformation has some caveats that must be considered. Since the logarithm function ($\log x$) results in real values only for $x > 0$, there must be a guarantee that the reference values are all positives and non-null.

One possibility is to calculate the references by using the absolute values of each element of the bicluster and, if the element $a_{i,j} = 0$, then a small $\epsilon > 0$ value is added. With this procedure it is possible to maintain the multiplicative coherence while using the same calculation of MSR as given by Eq. 4.

Some authors have already adapted some biclustering algorithms in order to deal with multiplicative coherence, but most of this work is to simply transform the data matrix, so that the multiplicative coherence would be changed to additive, and thus it would completely disregard any additive coherent bicluster [12].

Another approach to find multiplicative biclusters was proposed in [13] where it was devised another objective-function and the multiplicative coherence was found by means of a factor analysis. However, as in the previous approach, it was able to find only this kind of coherence.

Finally, in [14], it was proposed the Geometric bicluster, in which it was explored biclusters with linear dependencies. These linear dependencies could translate into a mix of an additive and multiplicative coherence, but this approach has some drawbacks: the increase of computational cost, the increase on the bicluster's complexity (for interpretability) and the focus on the search for a linear dependency may induce the algorithm to prefer an overfitted model with lower volume and lower MSR rather than a more general model with larger volume and larger MSR, and still below the acceptable threshold.

So, the aim of this paper is to adapt a biclustering method to be able to find the most proper coherent bicluster, being additive or multiplicative, for each portion of the dataset. As a consequence, the final set of biclusters contains both types of coherent biclusters that maximize the volume and minimizes the mean squared residue.

IV. SWARMBCLUSTER

The SwarmBCluster is a biclustering algorithm based on Swarm Intelligence [15], and proposed in [4]. This algorithm is able to find a set of biclusters that maximizes the coverage of the dataset without significantly degrading the average volume and the degree of coherence. This is an important feature, since most biclustering algorithms aim only to minimize the MSR and maximize the volume, generating overlapping biclusters that do not properly cover the dataset, which can lead to redundant information in some portions of the dataset and lack of information in other portions.

```

1: initialize a candidate list with every index of rows and columns in the
   dataset
2: create an empty bicluster list
3: while candidate list is not empty AND the bicluster list is smaller than
   bic_max do
4:   get a row index from the candidate list
5:   perform ACO with this row index as the start point
6:   if the resulting bicluster overlaps any other from the bicluster list
       more than over_thr % then
7:     if current bicluster is better than the overlapped bicluster then
8:       replace overlapped bicluster
9:     end if
10:  else
11:    insert bicluster into bicluster list
12:  end if
13:  update candidate list by removing the elements contained on the
     generated bicluster
14: end while

```

Fig. 3. Pseudo-code of the SwarmBcluster algorithm

Specifically, the SwarmBCluster is based on an Ant Colony Optimization (ACO) algorithm originally proposed by Dorigo [16], based on a study done in [17] on the behavior of Argentine ants when they are foraging for food.

Argentine ants seek food sources by randomly walking the region around the nest. When an ant finds a food source, it returns to the nest depositing a chemical substance called pheromone, thereby it leaves a pheromone trail from the food source discovered until the nest. Whenever an ant starts looking for food, it tends to follow a path with probability proportional to the intensity of pheromone. Obviously, the smaller tracks are privileged because they are traveled more quickly. Therefore, in the same interval, the pheromone trail of this track will be reinforced more often and, consequently, the effect of evaporation of pheromone is mitigated. Thus, as the shortest path tends to have a higher concentration of pheromone, it is likely that the ants will find and follow it. Because of the direct appeal, the first and most well-known application of ACO is for the Travelling Salesman Problem (TSP) [18].

The SwarmBCluster algorithm is illustrated in Fig. 3. This algorithm is composed of three main heuristics that interact with each other:

- a constructive heuristic, that starts with an initial bicluster with 1 row and m columns of an $n \times m$ dataset, and iteratively inserts rows and removes columns to control the residue;
- a dynamically built candidate list that supplies, to the first heuristic, the initial biclusters that are more likely to lead to final biclusters that present maximum coverage;
- the Ant Colony Optimization meta-heuristic (ACO) that is able to indicate, to the first heuristic, the best order in which the rows should be inserted in order to maximize the bicluster's volume.

Given an $n \times m$ dataset, the constructive heuristic starts with a bicluster composed of a single row and all the columns from this dataset. Then, this heuristic sequentially inserts rows of this dataset in a given order. After each insertion, it calculates the MSR for each column and removes those

with an MSR higher than a given threshold (δ). Whenever the bicluster reaches a minimum number of rows, this constructive procedure is stopped and a local search based on node insertion proposed in [3] is performed.

Since the order in which the rows are inserted influences the final results, the ACO is used at this step to look for the optimal order of rows that will lead to a bicluster with maximum volume. Since finding the optimal order of points from a giving set is equivalent to the Traveling Salesman Problem, it is straightforward to use this meta-heuristic in this situation. This procedure is further explained by the pseudo-algorithm in Fig. 4.

Another aspect of this constructive heuristic is that, depending on the initial bicluster, the final bicluster will contain a distinct set of rows and columns. So, if it is desirable to find a set of biclusters that maximizes the coverage of the dataset, it is appropriate to choose carefully the initial biclusters for the constructive heuristic so that the new biclusters contain a subset of rows and columns different from the previously found ones.

This task is performed by building a candidate list that enlists which elements of the data set have not yet been covered by any bicluster. The row with the highest number of columns not yet covered will be chosen as the next initial bicluster since it will most probably lead to an unexplored portion of the dataset.

So, in summary, the constructive heuristic is applied iteratively for a predefined number of times and, at each time, a different initial bicluster is provided by the candidate list and improved by successive insertions of rows. For each initial bicluster, the order in which the rows will be inserted is determined by an ACO algorithm.

Aiming at finding both additive and multiplicative biclusters, the SwarmBCluster requires a modification during the ACO procedure, as illustrated by the pseudo-algorithm in Fig. 4. The modification is depicted in lines 4 and 5: after finding the additive coherent bicluster, as default, the algorithm repeats the same heuristic procedure to find a multiplicative coherent bicluster, from the same starting point, but now using the transformed representative data briefly described in Section III-A. With both biclusters built, and with the guarantee of having an MSR lower than the specified δ , the one with a larger volume is chosen as the final solution built by the given ant.

The data transformation is done during the creation of the bicluster. Whenever a new row or column is inserted, the natural logarithm of the absolute value (or of a small ϵ if the value is 0) of each element of this row/column is stored on the bicluster data structure, instead of their actual values. In order to calculate the multiplicative coherent MSR, while still being able to compare with an MSR associated with an additive coherent bicluster, Eq. 5 must be modified to:

$$r_{i,j} = |a_{i,j}| - e^{la_{I,j} - la_{i,j} + la_{I,j}}, \quad (6)$$

where $|\cdot|$ is the absolute value function, $a_{i,j}$ is the value

```

1: for  $it$  from 1 to  $max\_it$  do
2:   for  $ant$  from 1 to  $max\_ants$  do
3:     build an additive coherent bicluster with the given starting point
4:     build a multiplicative coherent bicluster with the given starting point
5:     the bicluster with higher volume is chosen as the ant solution
6:   end for
7:   update the pheromone
8: end for
9: returns the best bicluster found

```

Fig. 4. Pseudo-code to the ACO algorithm when building a bicluster

of element (i, j) without the logarithm transformation and $la_{I,j}, la_{i,J}, la_{I,J}$ are, respectively, the average values of column j , row i and of the entire log-transformed bicluster. With this modified equation it is possible to compare the unchanged value of the dataset with the value expected by the bicluster model, leading to an error of the same magnitude obtained by using Eq. 5 for additive coherence.

It is worth noticing that, since the portion of the dataset explored is influenced by the candidate list, this simple procedure helps finding out what type of coherence is best for each part of the dataset.

Because of this particular characteristic of SwarmBcluster, it is not possible to compare the results obtained with such approach with any other approach from the literature, even if they are modified with the data transformation proposed here. Most population-based algorithms specialized on finding biclusters keep in the population only the fittest biclusters, according to their MSR and Volume, but regardless of the region of the dataset they represent. This creates a tendency of exploring just part of the dataset, in contrast with SwarmBcluster that only compares two biclusters from a similar portion and tries to maximize the dataset coverage. This characteristic of such methods was verified and further analyzed in [11], and the advantage of SwarmBcluster under these circumstances was verified on the experimental results realized in [4].

Regarding the increase in complexity, the computational cost of this algorithm will be increased proportionally to the number of types of coherence analyzed. Finally, this procedure can be extended to find other types of bicluster like constant-valued, constant rows, constant columns, or any other coherence measure that can be adopted by the constructive heuristic.

V. EXPERIMENTS

This section will describe the experimental set adopted to compare the effect of finding multiplicative coherent biclusters together with additive coherence within two datasets from the literature.

A. Methodology

In order to properly analyze and assess the usefulness of finding additive and multiplicative coherent biclusters at the same time, three experiments will be evaluated with two different datasets. The experiments consist of finding three set of biclusters, each one containing 100 biclusters: one composed only of biclusters with additive coherence,

TABLE I
MEAN AND STANDARD DEVIATION VALUES OF THE MEAN-SQUARED RESIDUE AND AVERAGE VOLUME OF THE ADDITIVE, MULTIPLICATIVE AND MIXED COHERENT BICLUSTERS OBTAINED WITH SWARMBCLUSTER FOR THE YEAST DATASET.

	MSR	Volume
Additive	148.94 ± 4.25	4154.26 ± 201.32
Multiplicative	150.71 ± 4.93	3086.21 ± 81.16
Mixed	167.84 ± 5.10	4538.02 ± 103.54

other only with multiplicative coherence and one with mixed coherence types. The datasets that will be used here are the well known *Yeast* and *Human* data that can be found at <http://arep.med.harvard.edu/biclustering>.

The *Saccharomyces cerevisiae* cell cycle expression data from Cho *et al.* [19], also known as the *Yeast* dataset, contains 2,884 genes under 17 conditions, and it was adapted in this work according to Cheng & Church [3], with the null values being replaced by random numbers in the range [0, 600].

The *Human B-Cell Lymphoma* expression data, first employed in [20], and hereby known as the *Human* dataset, has 4,026 genes under 96 conditions, and the expression levels were treated in the same way as the *Yeast* dataset, with the few missing values being replaced by random numbers in the range [−750, 650]. As suggested in [3], the δ constraint used for each dataset was 300 for the *Yeast* and 1200 for the *Human*.

The parameters for the SwarmBcluster were set similarly as suggested in [4] in order to balance the computational performance and the quality of results. Since the algorithm runs the ACO procedure multiple times in order to obtain a high coverage of the dataset, the number of ants and iterations were reduced on both experiments. For the common setup the pheromone decay parameter was set to 0.2, the maximum overlap allowed was 60% on both datasets and the number of ants were set to 5 within 5 iterations. For the *Yeast* dataset the minimum number of columns on a bicluster was set to 7 and 20 for the *Human* dataset.

The experiments were all performed on an Phenom 2 X4 955 @ 3.20GHz with 2GB of RAM, each experiment using just a single core.

B. Results

1) *Yeast Microarray Dataset*: from Table I it is possible to see that when comparing the sets of additive and multiplicative biclusters, the additive set has, on average, 35% larger biclusters while maintaining an equivalent MSR. This indicates that, in this dataset, the occurrence of additive coherence is more frequent than multiplicative. But, within the mixed set, the average bicluster volume was 9% and 47% larger than the additive and multiplicative, respectively, and the MSR raised just 12% and 11%. This indicates that, although there are more additive coherence inside the dataset, some portions still have a predominance of multiplicative coherence.

This can be confirmed by the percentage of multiplicative biclusters contained on the mixed set. There was an average of

TABLE II

MEAN AND STANDARD DEVIATION VALUES OF THE TIME TAKEN, IN SECONDS, TO FIND A SET OF ADDITIVE, MULTIPLICATIVE AND MIXED COHERENT BICLUSTERS WITH SWARMBCLUSTER FOR THE YEAST DATASET.

	Time (sec.)
Additive	19.83 ± 0.70
Multiplicative	18.18 ± 1.57
Mixed	34.70 ± 4.52

TABLE III

MEAN AND STANDARD DEVIATION VALUES OF THE MEAN-SQUARED RESIDUE AND AVERAGE VOLUME OF THE ADDITIVE, MULTIPLICATIVE AND MIXED COHERENT BICLUSTERS OBTAINED WITH SWARMBCLUSTER FOR THE HUMAN DATASET.

	MSR	Volume
Additive	632.56 ± 11.51	4355.98 ± 218.12
Multiplicative	442.01 ± 5.13	25193.90 ± 813.62
Mixed	444.45 ± 4.61	25554.90 ± 816.35

31% of them on this set, so that, the simultaneous discovery of multiplicative coherence plays a significant role in the biclustering process.

Table II reports the time, in seconds, taken to find each set of biclusters. These values confirm the increase in computational cost by searching for multiple coherences. While searching for either additive or multiplicative biclusters takes about the same time, when looking for both simultaneously almost doubles the time needed.

2) *Human Microarray Dataset*: now considering the Human dataset, as reported on Table III, this dataset most likely contains much better multiplicative coherent biclusters than additive. This is most noticeable by the average volume, in which, an increase of 480% was obtained while the MSR has dropped by a rate of 30% when compared to the additive coherent set. Since 100% of the biclusters from the mixed set are from the multiplicative type, the results obtained are statistically equivalent to the multiplicative set.

Given that the average volume of the multiplicative biclusters are much larger than the additive, it would be possible, if desired, to lower the δ constraint in order to find smaller biclusters but with a lower MSR. With a $\delta = 450$ it was possible to maintain the average volume at 1424.03, closer to the additive coherent set, while the mean MSR was at 212.50, a reduction of approximately 50%.

About the computational time, in order to find the additive and multiplicative sets, it took around 80 and 60 seconds respectively, while to find the mixed set it took 138 seconds, approximately the sum of the time taken by the two individual sets, thus confirming that the complexity is proportional to the number of coherences considered by the algorithm.

3) *Comparison with the literature*: as a final experiment, the results obtained by SwarmBcluster were compared with another algorithm from the literature, called MOEABIC [21]. This algorithm was reported to have the closest results to

TABLE IV

MEAN AND STANDARD DEVIATION VALUES OF THE TIME TAKEN, IN SECONDS, TO FIND A SET OF ADDITIVE, MULTIPLICATIVE AND MIXED COHERENT BICLUSTERS WITH SWARMBCLUSTER FOR THE HUMAN DATASET.

	Time (sec.)
Additive	82.25 ± 4.45
Multiplicative	60.09 ± 3.21
Mixed	138.33 ± 5.22

TABLE V

MEAN VALUES OF THE MEAN-SQUARED RESIDUE, AVERAGE VOLUME AND COVERAGE OF THE ADDITIVE, MULTIPLICATIVE AND MIXED COHERENT BICLUSTERS OBTAINED WITH SWARMBCLUSTER AND MOEABIC FOR THE YEAST DATASET.

	SwarmBcluster			MOEABIC	
	Add.	Mult.	Mix.	Add.	Mult.
MSR	148.94	150.71	167.84	288.62	300.00
Volume	4154.26	3086.21	4538.02	4800.40	3569.52
Coverage	0.79	0.78	0.80	0.51	0.48

those obtained by the original SwarmBcluster [4]. Since this algorithm evolves every bicluster from the set in parallel, it is unable to find a mixed set of biclusters, so it will be tested two versions: the original version with additive coherence and a modified version with the calculation of multiplicative coherence as described in this paper. Also, the data coverage of the bicluster set for each algorithm will be reported, to measure the redundancy of each set.

In Table V, concerning the Yeast dataset, it is possible to see that the results obtained by SwarmBcluster had a smaller volume than those obtained by MOEABIC but with significantly lower MSR and a much higher coverage. The results obtained on the Human dataset, depicted in Table VI, shows that SwarmBcluster was capable of outperforming MOEABIC in every aspect, except on the average volume of the additive biclusters set. These results illustrate the main advantage of SwarmBcluster over other approaches: the ability to find a more diverse set of biclusters and correctly identify which type of coherence is more suitable for a given portion of the dataset.

VI. CONCLUSION

In this paper a Swarm Intelligence approach to solve the δ -bicluster problem, called SwarmBcluster, was modified to be able to find both additive and multiplicative coherent biclusters concurrently. This problem has the objective to find, inside a dataset, a set of biclusters with Mean Squared Residue lower than or equal to a given δ . There are several aspects that can be used to measure the quality of a bicluster, for example, the maximization of its volume and the minimization of the mean squared residue.

The mean squared residue is a calculation that measures how far a given bicluster is from a coherent model. This coherent model describes what type of similar behavior the elements of the bicluster have. One of this behavior is called

TABLE VI

MEAN VALUES OF THE MEAN-SQUARED RESIDUE, AVERAGE VOLUME AND COVERAGE OF THE ADDITIVE, MULTIPLICATIVE AND MIXED COHERENT BICLUSTERS OBTAINED WITH SWARMBCLUSTER AND MOEABIC FOR THE HUMAN DATASET.

	SwarmBcluster			MOEABIC	
	Add.	Mult.	Mix.	Add.	Mult.
MSR	632.56	442.01	444.45	1052.90	958.41
Volume	4355.98	25193.90	25554.90	5552.11	6152.42
Coverage	0.26	0.58	0.58	0.21	0.48

coherence and may refer to several types of relation between elements. Two coherences of special interest are the additive and multiplicative ones, being the additive the most studied on biclustering applications.

Since the type of correlation among elements from a given portion of a dataset is unknown, it is usually assumed one single coherence for the whole dataset. In this paper, though, it was shown that the SwarmBcluster algorithm is capable of handling more than one type of coherence at the same time while inspecting which type is better on a given portion of the dataset. The two studied coherence measures were the additive and the multiplicative coherences.

The experiments showed that the ability to automatically choose the coherence type can improve the quality of the biclusters found by the algorithm without increasing the computational cost too much. As further steps of the research, it will be investigated the inclusion of other types of coherence, and a thorough comparison with biclustering algorithms for each type of coherence from the literature will be performed.

REFERENCES

- [1] A. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.
- [2] F. O. de França, G. Bezerra, and F. J. Von Zuben, "New Perspectives for the Biclustering Problem," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, Vancouver, Canada, 2006, pp. 753–760.
- [3] Y. Cheng and G. M. Church, "Biclustering of expression data," in *Proc. of the 8th Int. Conf. on Intelligent Systems for Molecular Biology*, 2000, pp. 93–103.
- [4] F. O. de França and F. J. Von Zuben, "Finding a high coverage set of δ -biclusters with swarm intelligence," in *Evolutionary Computation (CEC), 2010 IEEE Congress on*, Barcelona, Spain, jul. 2010, pp. 2523–2530.
- [5] R. Agrawal, J. Gehrke, D. Gunopulus, and P. Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications," in *Proc. of the ACM/SIGMOD Int. Conference on Management of Data*, 1998, pp. 94–105.
- [6] P. A. D. de Castro, F. O. de França, H. M. Ferreira, and F. J. Von Zuben, "Applying Biclustering to Text Mining: An Immune-Inspired Approach," in *Artificial Immune Systems, Proc. of the 6th International Conference on Artificial Immune Systems (ICARIS)*, ser. Lecture Notes in Computer Science, L. N. de Castro, F. J. Von Zuben, and H. Knidel, Eds., vol. 4628, Santos, Brazil, 2007, pp. 83–94.
- [7] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *Proc. of the 7th Int. Conf. on Knowledge Discovery and Data Mining*, 2001, pp. 269–274.
- [8] J. A. Hartigan, "Direct clustering of a data matrix," *Journal of the American Statistical Association (JASA)*, vol. 67, no. 337, pp. 123–129, 1972.
- [9] P. A. D. de Castro, F. O. de França, H. M. Ferreira, and F. J. Von Zuben, "Applying Biclustering to Perform Collaborative Filtering," in *Proc. of the 7th International Conference on Intelligent Systems Design and Applications*, Rio de Janeiro, Brazil, 2007, pp. 421–426.
- [10] C. Tang, L. Zhang, I. Zhang, and M. Ramanathan, "Interrelated two-way clustering: an unsupervised approach for gene expression data analysis," in *Proc. of the 2nd IEEE Int. Symposium on Bioinformatics and Bioengineering*, 2001, pp. 41–48.
- [11] G. P. Coelho, F. O. de França, and F. J. Von Zuben, "Multi-objective biclustering: When non-dominated solutions are not enough," *Journal of Mathematical Modelling and Algorithms*, vol. 8, no. 2, pp. 175–202, 2009.
- [12] K. Cheng, N. Law, W. Siu, and T. Lau, "BiVisu: software tool for bicluster detection and visualization," *Bioinformatics*, vol. 23, no. 17, p. 2342, 2007.
- [13] S. Hochreiter, U. Bodenhofer, M. Heusel, A. Mayr, A. Mitterecker, A. Kasim, T. Khamiakova, S. Van Sanden, D. Lin, W. Talloen *et al.*, "FABIA: factor analysis for bicluster acquisition," *Bioinformatics*, vol. 26, no. 12, p. 1520, 2010.
- [14] X. Gan, A. Liew, and H. Yan, "Biclustering gene expression data based on a high dimensional geometric method," in *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, vol. 6. IEEE, 2005, pp. 3388–3393.
- [15] J. Kennedy, R. Eberhart, and Y. Shi, *Swarm intelligence*. Springer, 2001.
- [16] M. Dorigo, E. Bonabeau, G. Theraulaz *et al.*, "Ant algorithms and stigmergy," *Future Gener Comput Syst*, vol. 16, no. 8, pp. 851–871, 2000.
- [17] S. Goss, S. Aron, J. Deneubourg, and J. Pasteels, "Self-organized shortcuts in the Argentine ant," *Naturwissenschaften*, vol. 76, no. 12, pp. 579–581, 1989.
- [18] E. Lawler, *The traveling salesman problem: a guided tour of combinatorial optimization*. John Wiley & Sons Inc, 1985.
- [19] R. Cho, M. Campbell, E. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T. Wolfsberg, A. Gabrielian, D. Landsman, D. Lockhart, and R. Davis, "A genome-wide transcriptional analysis of the mitotic cell cycle," *Molecular Cell*, vol. 2, pp. 65–73, 1998.
- [20] A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, J. I. Powell, L. Yang, G. E. Marti, T. Moore, J. Hudson Jr., L. Lu, D. B. Lewis, R. Tibshirani, G. Sherlock, W. C. Chan, T. C. Greiner, D. D. Weisenburger, J. O. Armitage, R. Warnke, R. Levy, W. Wilson, M. R. Grever, J. C. Byrd, D. Botstein, P. O. Brown, and L. M. Staudt, "Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling," *Nature*, vol. 403, pp. 503–510, 2000.
- [21] S. Mitra and H. Banka, "Multi-objective evolutionary biclustering of gene expression data," *Pattern Recognition*, vol. 39, pp. 2464–2477, 2006.