Text Document Classification Using Swarm Intelligence

André L. Vizine[1], Leandro N. de Castro[1] & Ricardo R. Gudwin[2]
{vizine,lnunes}@unisantos.br; gudwin@dca.fee.unicamp.br
[1]Catholic University of Santos, R. Dr. Carvalho de Mendonça, 144, 11070-906, Santos/SP – Brazil;
[2]State University of Campinas, C.P. 6101, 13083-852, Campinas/SP – Brazil.

Abstract – *This paper presents an algorithm for the automatic grouping of PDF documents, and with potential application for Web document classification. The algorithm developed is based on an ant-clustering algorithm, which was inspired by the behavior of some ant species in the organization their nests. To apply the ant-clustering algorithm for text document classification, two modifications had to be introduced in the standard algorithm: 1) the use of a metric to evaluate the similarity degree of text data, instead of numeric data; and 2) the proposal of a cooling schedule for a user-defined parameter so as to improve the convergence properties of the algorithm. To illustrate the behavior of the modified algorithm, it was applied to sets of real-world documents taken from the IEEE WCCI -1998 CD.*

1. Introduction

Until very recently, obtaining information about a given subject involved going to a library or university and searching for the desired contents. All bibliographical resources of a library (e.g., books, journals, magazines and newspapers) are grouped by indices, i.e., collections of terms that point to the sites where they can be found. These terms can be the names of the authors, the subjects, the year of publication, and so forth. With time, not only the number of libraries increases, but also the amount of information available. To minimize this problem, information retrieval systems have been developed and widely used in libraries, universities, companies and all other places where information resources have to be stored and consulted. Information retrieval systems are aimed at helping the storage of new information resources and speeding up the search for a specific subject [1],[2].

The Internet has emerged as one of the most important information resources, in most cases of public use, available nowadays. This can be easily observed by the broad number of digital libraries in the Web [3]. As is the case with the "physical" libraries, digital or virtual libraries also suffer from the difficulties in organizing and searching for information. Although information retrieval systems have contributed significantly to the organization and retrieval of information, the success of the system still depends on maintenance, because one has to be responsible for taking the new information resources, indexing and cataloguing them. These processes are tedious and time consuming.

This paper presents a system for the automatic organization of digital documents in PDF format. The approach is based on an ant-clustering algorithm proposed by Lumer and Faieta [4] as a development of the ideas introduced by Deneubourg et al. [5]. This method was designed as part of an academic virtual community currently under development. This community is characterized as a scientific paper collection (PDF files) automatically classified and stored in folder structures of a server and in which academics are able to exchange experience and knowledge.

This paper is organized as follows. Section 2 briefly introduces swarm intelligence and the ant-clustering algorithm. Section 3 describes how the algorithm was implemented to solve text clustering, and Section 4 presents some simulation results. Conclusions and some proposals for future works are provided in Section 5.

2. Swarms, Ants and Clustering

The social behavior of ants has attracted the interest of researchers in many different disciplines, from the biosciences to computer science and engineering. One emerging field of investigation that has been increasingly receiving attention over the past years is the so-called biologically inspired computing [6], in particular, swarm intelligence [7],[8]. The term swarm intelligence was coined in the late 1980's to refer to cellular robotic systems in which a collection of simple agents in an environment interact according to local rules [9],[10].

Two main lines of research can be identified in swarm intelligence: 1) the works based on social insects [7]; and 2) the works based on the ability of human societies to process knowledge [8]. Although the resultant approaches are quite different in sequence of steps and sources of inspiration, they present some commonalities. In general terms, both rely on a population (colony or swarm) of individuals (social insects or particles) capable of interacting (directly or indirectly) with the environment and each other. As a result of these interactions there may be a change in the environment and/or in the individuals, what will lead to useful emergent phenomena.

Among the many social behaviors of ants, researchers have registered the way some ant species work collaboratively in the task of grouping dead bodies so as to keep the nest clean [11],[12]. After placing corpses of ants randomly in a certain environment, it can be observed that, with time, the ants tend to cluster all dead bodies in specific regions of the environment, thus forming piles of dead bodies. The first ant-clustering algorithm inspired by this clustering behavior of ants was introduced in [5], where a population of robots had to group together objects without any central control.

Lumer and Faieta [12] adapted the robots ant-clustering algorithm for the analysis and classification of numerical data, thus introducing the standard ant-clustering algorithm (ACA). Since its proposal, in 1994, the ACA has passed through some modifications and has been applied to several domains, from data mining [12], to graph-partitioning [13]-[15], to text-mining [16]-[18]. Independently of the application domain and particular version of the algorithm, ant-clustering algorithms based on ACA follow a set of basic, general principles.

Given an input data set composed of *N* *l*-dimensional vectors to be clustered, these data are spread all over a bi-dimensional (toroidal) grid of size $m \times m$. Actually, the data themselves are not spread over the grid, only some sort of indices that indicate where a given object is placed. A colony of ants (agents) is allowed to move on the grid picking up, carrying and dropping off objects based on some probabilistic rules. The movement of an ant is characterized by its displacement in a grid cell in any direction adjacent to its current position. The ants can perceive a neighborhood in the environment, the most common one being a square neighborhood of size 3×3. In the beginning of the iterative process of adaptation, objects and ants are randomly placed on the grid.

The ants then start moving randomly on the grid. If an ant is not carrying an object and finds an object *i* in its neighborhood, it picks up this object with a probability that is inversely proportional to the number of similar objects in the neighborhood, as described in Eq. (1).

$$P_{pick}(i) = \left(\frac{k_p}{(k_p + f(i))^2} \right) \qquad (1)$$

If, however, the ant is carrying an object *i* and perceives a neighbor cell in which there are other objects, then the ant drops off the object it is carrying, with a probability that is directly proportional to the object's similarity with the perceived ones, as described in Eq. (2).

$$P_{drop}(i) = \left(\frac{f(i)}{(k_d + f(i))^2} \right) \qquad (2)$$

The parameters k_p and k_d are, respectively, the picking and dropping constants that weighs the influence of the function *f*(*i*) on the picking and dropping probabilities. Function *f*(*i*) provides an estimate of the density and similarity of elements in the neighborhood of object *i*. The pseudocode presented in Algorithm 1 summarizes the standard ant-clustering algorithm (ACA).

```
procedure ACA (max_it,$k_p$,$k_d$)
   place every item $i$ on a random cell of the grid
   place every ant $k$ on a random cell of the grid unoccupied by ants
   t ← 1
   while  t < max_it do,
      for $i$ = 1 to $N$ do,   // for every ant
          if  unladen ant AND cell occupied by item $\mathbf{x}_i$, then
              compute $f(\mathbf{x}_i)$ and $p_p(\mathbf{x}_i)$
              pick up item $\mathbf{x}_i$ with probability $p_p(\mathbf{x}_i)$
          else if ant carrying item $\mathbf{x}_i$ AND cell empty, then
                     compute $f(\mathbf{x}_i)$ and $p_d(\mathbf{x}_i)$
                     drop item $\mathbf{x}_i$ with probability $p_d(\mathbf{x}_i)$
          end if
          move to a randomly selected neighboring and unoccupied cell
       end for
       t ← t + 1
   end while
   print location of items
end procedure
```

Algorithm 1: Standard ant-clustering algorithm.

The authors [4] suggested the following function *f*(*i*) as the local density of items in the neighborhood of object *i*:

$$f(i) = \begin{cases} \dfrac{1}{s^2} \displaystyle\sum_{j \in \mathrm{Neigh}_{(s \times s)}(i)} \left[1 - \dfrac{d(i,j)}{\alpha} \right] & \text{if } f > 0 \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

where *f*(*i*) is a measure of the average similarity of object *i* with another object *j* in the neighborhood of *i*, α is a factor that defines the scale of dissimilarity, and *d*(*i*,*j*) is the distance between two items in their original *l*-dimensional space. Parameter α determines when two items should or should not be located next to each other. For instance, if α is too large, there is not much discrimination between two items, leading to the formation of clusters composed of items that should not belong to the same cluster; and vice-versa.

3. Ant Algorithm for Text Clustering

To cluster PDF files, it is first necessary to convert them into text documents. Then, they have to be transformed into collections of words that will represent an object on the grid. This transformation is automatically obtained through the calculation of the relative frequency of a word in the documents (Eq. (4)). Let $f_j(w)$ corresponds to the number of times word w appears in document j, i.e., the frequency of word w in document j. Then, $F_j(w)$ represents the relative frequency of word w in all documents:

$$F_j(w) = \frac{f_j(w)}{\sum_v f_j(v)}; \quad v \neq w \tag{4}$$

Note that $0 < F_j(w) < 1$ and $\Sigma_w F_j(w) = 1$. This normalization serves the purpose of disregarding the number of words in the document and, instead, measure the relative importance of a word compared to the other words contained in the same document.

Instead of using the Euclidean distance as a measure of dissimilarity, the cosine measure [19], which determines the similarity between two vectors independently of their magnitude, is going to be used. One vector represents the set of words extracted from the document being carried by the ant, and the other vector represents the collection of words extracted from the documents in the neighborhood of the ant. Eq. (5) returns the cosine of the angle between these two vectors. The cosine is equal to 1 when the vectors point in the same direction, and zero when they form a 90 degrees angle.

$$sim(D_D, D_Q) = \frac{\sum_{k=1}^{N} F_{Dk} F_{Qk}}{\sqrt{\sum_{k=1}^{N} {F_{Dk}}^2 \sum_{k=1}^{N} {F_{Qk}}^2}}, \tag{5}$$

where F_{Dk} is the frequency of word k in the set of words extracted from the document being carried by ant D, and F_{Qk} is the relative frequency of word k in document Q that exists in the neighborhood of the ant.

Thus, each document is transformed into an object whose structure is an l-dimensional vector, $x_1x_2...x_l$ which corresponds the relative frequencies of the relevant words extracted. Table I illustrates the representation of each object.

Table I – Objects that represent documents.

Object	x_1	...	x_l
O_1	0.123	…	0.232
…	…	…	…
O_N	0.012	…	0.156

After the generation of the objects, the ant-clustering algorithm described previously is applied. The only difference is the calculation of $f(i)$, where the cosine measure $(sim(\cdot,\cdot)/\alpha)$, is used instead of $(1 - d(\cdot,\cdot)/\alpha)$.

After some preliminary tests, it was noticed that the algorithm could never converge to a stable configuration of the grid; that is, the ants were constantly building and destructing clusters. Therefore, one form of modifying ACA in order to promote a stabilization of the grid had to be proposed. The approach adopted here corresponds to gradually cooling down the value of parameter k_p so as to reduce the probability of an ant picking up an object after a certain number of iteration steps have passed. With this simple modification, the stopping criterion of the algorithm becomes either a maximum number of cycles (1cycle = 10,000 steps of each ant) or a minimum value for k_p. In both cases, the chosen value has to be such that ants are no longer picking up objects from the grid, thus resulting in a final, stable clustering solution.

4. Performance Evaluation

In order to assess the applicability of the modified algorithm, called here ACA*, for text document clustering, a benchmark dataset was used. Sets of PDF files were copied from the WCCI – IEEE 1998 World Congress on Computational Intelligence: Proceedings of IJCNN 1998, FUZZ-IEEE 1998 and ICEC 1998.

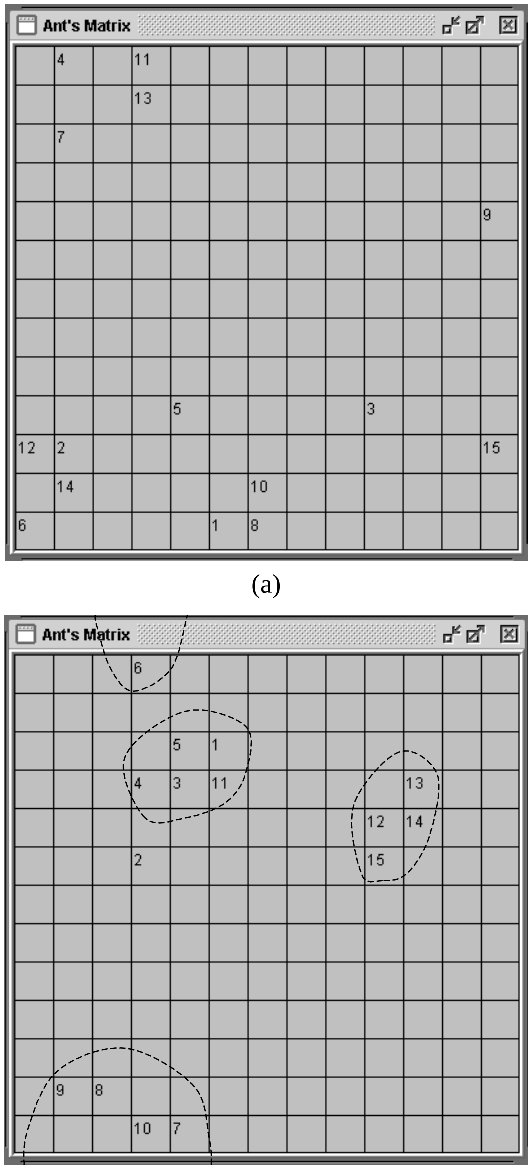
Initially, the ACA* was tested using 15 articles of three distinct groups (areas): evolutionary computation (EC), artificial neural networks (ANN) and fuzzy systems (FS). Table II summarizes the information about each of them.

Table II – Indices of the documents used in the preliminary simulations.

Documents	Groups (Area)
1 to 5	EC
6 to 10	FS
11 to 15	ANN

The algorithm was run 20 times, with each execution corresponding to 30 cycles. The parameters used for running the algorithm were: $k_p = 0.01$ (initial value); $\alpha = 0.7$; $k_d = 0.06$; $k_{pmin} = 0.001$; $Nants = 1$; grid size: 13×13.

In these initial tests, two objects from the whole data set behaved abnormally. Object number 2, which according to the database is classified as a document related to EC, often appeared in the FS group (objects 6 to 10). The same behavior was observed in the object number 11, which is related with ANN, but sometimes appeared in the EC group. Figure 1 illustrates the initial and final configurations of the grid.



(a)

(b)

Figure 1 – Initial (a) and final (b) configurations of the grid.

Table III summarizes the results obtained in the preliminary tests, where the column labeled Error(%) corresponds to the percentage classification error; column *Nw* contains the number of objects incorrectly grouped; column *C* corresponds to the number of clusters determined in each experiment; and column *Ni* presents the number of objects not grouped with any of the existing clusters. The last column shows what happened with objects 2 and 11: if falling into the EC group, FS group, or isolated. After investigating the contents of objects 2 and 11, it was possible to note that the relative frequency of their words pointed them to other groups. For instance, object 2, which originally belongs to the EC group, is titled “Adjusting fuzzy partitions by genetic algorithms and histograms for pattern classification problems”, and presents 96 occurrences of the word fuzzy. Experiment 3, from Table III, corresponds to the grid presented in Figure 1(b).

Table III – Simulation results obtained for the initial experiments. *Nw*: number of wrong items; *C*: number of clusters; *Ni*: number of isolated objects.

Experiment	**% Error**	***Nw***	***C***	***Ni***	**Objects 2, 11**
1	-		4		11(EC) 2(FS)
2	-		3		11(EC) 2(FS)

3	-		3	1	11(EC)
					2(isolated)

4	-		4	1	11(EC) 2(FS)

5	-		3	1	11(isolated)
					2(FS)

6	-		3	2	11(isolated)
					2(FS)

7	-		4	1	11(EC) 2(FS)

8	-		4	1	11(isolated)
					2(EC)

9	-		5		11(EC) 2(FS)
10	-		3		11(EC) 2(FS)
11	6.6	1	3	1	11(EC) 2(FS)

12	-		4	1	11(isolated)
					2(FS)

13	-		4	1	11(EC) 2(EC)

14	13.3	2	3	1	11(isolated)
					2(ANN)

15	13.3	2	3		11(EC)2(FS)
16	-		4		11(EC) 2(FS)
17	-		3	1	11(EC) 2(FS)
18	-		3	1	11(EC) 2(FS)
19	-		3	2	11(EC) 2(FS)
20	-		4	2	11(EC) 2(FS)
Mean **(Std)**	**4.63** **(18.02)**	**0.25** **(1.24)**	**3.5** **(1.56)**	**0.85** **(1.42)**	

Motivated by the encouraging results obtained with a small sample of documents, we took a larger number of objects to evaluate the algorithm. Out of a total of 1,151 PDF files on the WCCI 1998 CD, we randomly chose 90 documents for the next experiments. The reasons why choosing only 90 out of 1,151 files were twofold: i) the high computational cost necessary for running the algorithm for the whole data set; and ii) the grid size required to accommodate all the objects. In the present experiment, 30 objects from each group (EC, ANN, FS) were taken, and the following parameters used: $k_p = 0.01$ (initial value); grid size: 30×30; $k_d = 0.06$; $k_{pmin} = 0.001$; $\alpha = 0.7$; $Nants = 10$. Note that the only parameters that changed in relation to the previous experiments were the number of ants and the grid size.

Figure 2 illustrates one of the results obtained for this experiment. In this case, objects 0 to 29 belong to the EC group, objects 30 to 59 belong to the FS group, and objects 60 to 89 belong to the ANN group. Nine different clusters were determined by the ACA* algorithm, and only four objects were left isolated: 6, 10, 14, and 42. By investigating the isolated objects we observed the following: object 6 could not be appropriately converted from PDF to text file; objects 14 and 42 deal with, respectively, extracting rules from fuzzy neural networks and the use of fuzzy clustering; and object 10 proposed the development of an artificial brain based on the evolution of neural networks.

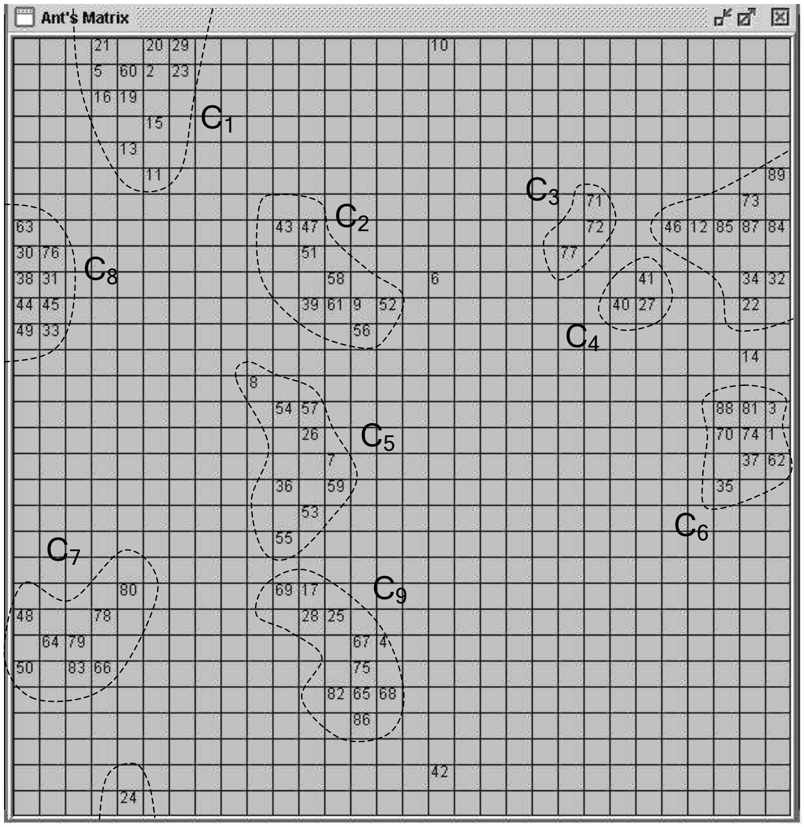


Figure 2 – Simulation result for the data set containing 90 objects.

Table IV summarizes the results presented in Figure 2.

Table IV – Simulation results for the data set containing 90 documents. *No*: number of objects in the cluster; *Nw*: number of objects classified incorrectly.

Cluster label	*Group*	*No*	*Nw*
C_1	EC	13	1
C_2	FS	9	2
C_3	ANN	3	0
C_4	FS	3	1
C_5	FS	9	3
C_6	ANN	9	4
C_7	ANN	8	2
C_8	FS	19	9
C_9	ANN	11	4

After investigating why some items are clustered in the wrong group, it could be observed that, although some documents do belong to one particular group, when the document is converted into an object for clustering, the context of the document is transformed into a collection of words with their respective relative frequencies. As an outcome, some words that are more characteristic of a certain group may appear too often in other groups as well. For instance, by taking object 4, titled "A spanning tree-based genetic algorithm for bicriteria topological network design", we noticed that the word "network" occurred 45 times in this document, and "network" is a word very characteristic of the ANN group.

5. Conclusions and Future Trends

This paper presented an algorithm for the automatic clustering of text documents using a classification technique inspired by the behavior of some ant species while organizing and cleaning their nests. As the ant-clustering algorithm used was originally developed to tackle numeric data, some modifications had to be introduced in order to adapt it to deal with text data. Furthermore, we also proposed a cooling schedule for a parameter of the algorithm resulting in the improvement of its convergence properties.

To evaluate the performance of the algorithm, real-world documents were extracted from the IEEE WCCI 1998 CD and presented to the algorithm. The system demonstrated to be capable of grouping together documents belonging to the same original group (fuzzy, neural or evolutionary). The cases the algorithm mixed a document corresponded to those cases in which the paper (PDF file) referred to hybrid approaches or in which the paper could be correctly classified into more than one group, thus leaving margin for a dual right classification. Furthermore, in many cases a document was inappropriately classified because it contained several words that were more common in a group than in its original group; for instance, the word network may appear several times in an EC paper.

There are several avenues for further investigation. An aspect that deserves particular attention is the misclassification due to the object (vector) generated from the words. An approach that can be used to tackle this problem is the automatic extraction of keywords from the documents [20] instead of taking all words. This would also reduce the associated computational cost. It is necessary to investigate the possibility of incorporating pheromone to the grid [21]; and implementing piles of objects [22], instead of placing one object per grid cell.

References

[1] R. Baeza-Yates and B. Ribeira-Neto, *Modern Information Retrieval*, Addison Wesley, (1999).

[2] S. Lawrence, C. L. Giles and K. Bolacker, "Indexing and Retrieval of Scientific Literature", *Proc. of the 8th Int. Conf. on Information and Knowledge Management*, pp. 139-146, (1999).

[3] S. Lawrence, C. L. Giles and K. Bollacker, "Research Feature Digital Libraries and Autonomous Citation Indexing", *IEEE Computer*, **32**(6), pp.67-71, (1999).

[4] E. Lumer and B. Faieta, "Diversity and Adaptation in Populations of Clustering Ants", *Proc. of the 3rd Int. Conf. on Simulation of Adaptive Behaviour: From Animals to Animats*, pp. 501-508. (1994).

[5] J. Deneubourg, N. Goss, N. Franks, A. Sendova-Franks, C. Detrain and L. Chrétien, "The Dynamics of Collective Sorting: Robot-Like Ant and Ant-Like Robot", *Proc. of the 1st Int. Conf. on Simulation of Adaptive Behavior: From Animals to Animats*, pp.356-365, (1991).

[6] L. N. de Castro, and F. J. Von Zuben, *Recent Developments in Biologically Inspired Computing*, Idea Group Inc., Hershey-PA, (2004).

[7] E. Bonabeau, M. Dorigo, M. and G. Théraulaz, *Swarm Intelligence from Natural to Artificial Systems*, Oxford University Press, (1999).

[8] J. Kennedy and R. Eberhart, *Swarm Intelligence*, Morgan Kaufmann Publishers, (2001).

[9] G. Beni, "The Concept of Cellular Robotic Systems", *Proc. of the IEEE Int. Symp. on Intelligent Control*, pp. 57-62, (1988).

[10] G. Beni, and J. Wang, "Swarm Intelligence", *Proc. of the 7th Annual Meeting of the Robotics Society of Japan*, pp. 425-428, (1989).

[11] E. Bonabeau, "From Classical Models of Morphogenesis to Agent-Based Models of Pattern Formation", *Artificial Life*, **3**, pp. 191-211, (1997).

[12] E. Lumer, and B. Faieta, "Exploratory Database Analysis via Self-Organization", *Unpublished Manuscript*, (1995).

[13] P. Kuntz and D. Snyers, "Emergent Colonization and Graph Partitioning, *Proc. of the 3rd Int. Conf. on Simulation of Adaptive Behaviour: From Animals to Animats*, pp. 494-500, (1994).

[14] P. Kuntz and D. Snyers, "New Results on an Ant-Based Heuristic for Highlighting the Organization of Large Graphs", *Proc. of the IEEE Congress on Evolutionary Computation*, pp. 1451-1458, (1999).

[15] P. Kuntz, D. Snyers, and P. Layzell, "A Stochastic Heuristic for Visualizing Graph Clusters in a Bi-Dimensional Space Prior to Partitioning", *Journal of Heuristics*, **5**(3), pp. 327-351, (1998).

[16] K. Hoe, W. Lai, and T. Tai, "Homogeneous Ants for Web Document Similarity Modeling and Categorization", *Proc. of the 3rd Int. Workshop on Ant Algorithms*, Lecture Notes in Computer Science 2463, Springer-Verlag, pp. 256-261, (2002).

[17] J. Handl, and B. Meyer, "Improved Ant-Based Clustering Sorting in a Document Retrieval Interface", *Proc. of the 7th International Conference on Parallel Problem Solving from Nature*, Lecture Notes in Computer Science 2439, Springer-Verlag, pp. 913-923, (2002).

[18] V. Ramos and J. J. Merelo, "Self-Organized Stigmergic Document Maps: Environments as a Mechanism for Context Learning", *Proc. of the 1st Spanish Conference on Evolutionary and Bio-Inspired Algorithms*, pp. 284-293, (2002).

[19] G. Salton and M. J. McGill, "The SMART and SIRE Experimental Retrieval Systems", In *Readings in Information Retrieval*, K. S. Jones and P. Willett (Eds.), Morgan Kaufmann Publishers Inc. (1997), pp. 381-399.

[20] K. Lagus and S. Kaski, "Keyword Selection Method for Characterizing Text Documents Maps", *Artificial Neural Networks*, **7**, pp. 371-376, (1999).

[21] V. Sherafat, L. N. de Castro, and E. R. Hruschka, "The Influence of Pheromone and Adaptive Vision on the Standard Ant Clustering Algorithm", In: L. N. de Castro and F. J. Von Zuben, *Recent Developments in Biologically Inspired Computing*, Chapter IX, pp. 207-234. Idea Group Inc., (2004).

[22] N. Monmarché, M. Slimane, G. Venturini, "On Improving Clustering in Numerical Databases with Artificial Ants", *Advances in Artificial Life*, D. Floreano, J. D. Nicoud, and F. Mondala (Eds.), *Lecture Notes in Computer Science 1674*, pp. 626-635, Springer-Verlag, Berlin, (1999).