IJIE – Iberoamerican Journal of Industrial Engineering
Periódico da área de Engenharia Industrial e áreas correlatas
Editor responsável: Nelson Casarotto Filho, Prof. Dr.
Organização responsável: Universidade Federal de Santa Catarina (UFSC)
Processo de avaliação de artigos por pares
Periodicidade: Semestral
Florianópolis, SC, v. 3, n. 1, p.184-197, Julho, 2011.
Artigo recebido em 08/08/2010 e aceito para publicação em 16/05/2011.

PERFORMANCE ANALYSIS OF A PROPOSED ANT-BASED CLUSTERING ALGORITHM

ANÁLISE DO DESEMPENHO DE UMA PROPOSTA DE UM ALGORITMO DE AGRUPAMENTO BASEADO EM FORMIGAS

Rosangela Villwock

Universidade Estadual do Oeste do Paraná
Doutora em Métodos Numéricos em Engenharia
Departamento de Matemática, UNIOESTE, PR, Brasil.
rosangelamat@hotmail.com

Maria Teresinha Arns Steiner

Universidade Federal do Paraná; Pontifícia Universidade Católica do Paraná
Doutora em Engenharia de Produção pela Universidade Federal de Santa Catarina
Programas de Pós-Graduação em Métodos Numéricos em Engenharia (PPGMNE) e de Engenharia de Produção (PPGEP), Programa de Pós-Graduação em Engenharia de Produção e Sistemas (PPGEPS), CP:19081; CEP:81531-990; Curitiba, PR, Brasil.
tere@ufpr.br; maria.steiner@pucpr.br

RESUMO: No algoritmo de Agrupamento baseado em Formigas, os padrões são espalhados numa grade e a cada formiga é atribuído um padrão. As formigas são responsáveis por carregar, transportar e descarregar os padrões na grade. Após a convergência do algoritmo de agrupamento, a recuperação dos grupos é feita usando-se as posições dos padrões na grade. O objetivo do presente estudo foi avaliar o desempenho de um Algoritmo de Agrupamento baseado em Formigas Proposto (ACAP), comparativamente ao Algoritmo de Agrupamento baseado em Formigas Modificado (ACAM). As principais modificações foram: substituição do padrão carregado pela formiga caso este não seja descarregado em 100 iterações consecutivas; comparação da probabilidade de descarregar um padrão na posição de sorteio com a probabilidade de descarregar este padrão em sua posição atual; avaliação da probabilidade de descarregar um padrão em uma nova posição, caso o padrão não seja descarregado na posição de sorteio e sim numa posição vizinha. Para a avaliação do desempenho do algoritmo aqui proposto foram utilizados dois exemplos reais: ÍRIS e WINE. Os resultados mostram que o ACAP deste estudo foi melhor do que o ACAM para os dois exemplos.

Palavras-chave: Mineração de Dados. Agrupamento de Padrões. Metaheurística.

ABSTRACT: In the Ant-Based Clustering Algorithm, patterns are spread throughout a grid and each ant is assigned a pattern. The ants are responsible for picking, transporting and dropping patterns on the grid. After the clustering algorithm converges, cluster recovery is done by using the positions of patterns on the grid. The purpose with this study was to evaluate the performance of the Ant-based Clustering Algorithm Proposed (ACAP) compared to the Ant-based Clustering Algorithm – Modified version (ACAM). The major changes were: replacement of the pattern carried by an ant in case it was not dropped within 100 consecutive iterations, comparing the probability of dropping a pattern at a random position with the probability of dropping this pattern at its current position; evaluate the probability of dropping a pattern at a new position, if the pattern is not dropped at a random position, but at a neighboring position. To assess the performance of the algorithm thus proposed, two real examples were used: ÍRIS and WINE. The results show that the ACAP in this study was better than the ACAM for the two examples.

Keywords: Data Mining. Patterns Clustering. Metaheuristics.

1 INTRODUCTION

Clustering based on Ants was initially suggested by Deneubourg et al. (1991). In it, ants were represented as simple agents that moved randomly on a square grid. The patterns were scattered within this grid and could be picked, transported and dropped by the agents (ants). These operations are based on similarity and on the density of the patterns that were distributed in the agents' local vicinity; isolated patterns, or those that are surrounded by dissimilar ones are more likely to be picked and dropped in a neighborhood of similars.

Decisions to pick and drop patterns are adopted by probabilities P_{pick} and P_{drop} given by Equations (1) and (2) below, respectively.

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

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

In these equations, *f(i)* is an estimate of the fraction of patterns located in the neighborhood that are similar to an ant's current pattern, and k_p and k_d are real constants. In the work of Deneubourg *et al.* (1991), the authors used $k_p = 0.1$ and $k_d = 0.3$. In this paper, the authors obtained the estimate of *f*, via a short-term memory of each ant, where the content of the last cell of the analyzed grid is stored. This choice of the neighborhood function *f(i))* was primarily motivated by its ease of implementation by simple robots.

Lumer and Faieta (1994, apud HANDL; KNOWLES; DORIGO, 2006) introduced a number of modifications to the model that allowed the manipulation of numerical data and improved the quality of the solution, as well as the algorithm's convergence time. The idea was to define a measure of similarity or dissimilarity between patterns, since in the algorithm initially proposed objects were similar if the objects were identical and were dissimilar if the objects were not identical. In the mentioned work, for the first time appears the topographic mapping.

According to Vizine et al., 2005, the general idea with this algorithm is to have similar data in the original *n*-dimensional space in neighboring regions of the grid, this is, data that are neighbors in the grid indicate similar patterns in the original space.

In the work of Lumer and Faieta (1994), apud Handl; Knowles and Dorigo, 2006), the decision of picking patterns is based on probability P_{pick} given by equation (1) above and the decision of dropping patterns is based on probability P_{drop} given by Equation () below, where *f(i)* is given by Equation (4).

$$P_{drop} = \begin{cases} 2f & , \quad if \quad f(i) < k_d \\ 1 & , \quad if \quad f(i) \geq k_d \end{cases} \tag{3}$$

$$f(i) = \max\left\{0, \frac{1}{\sigma^2} \sum_{j \in L} \left[1 - \frac{d(i,j)}{\alpha}\right]\right\} \tag{4}$$

In equation 4, $d(i, j)$ is a function of dissimilarity between patterns i and j belonging to the interval [0, 1]; α is a scalar parameter dependent on the data (patterns) belonging to the interval [0, 1]; L is the local neighborhood of size equal to σ^2, where σ is the perception radius (or neighborhood). In their work the authors used $k_p = 0.1$, $k_d = 0.15$ and $\alpha = 0.5$.

Ant-based Clustering Algorithms are inspired mainly in the versions proposed by Deneubourg et al. (1991), and Lumer and Faieta (1994), apud HANDL; KNOWLES; DORIGO (2006). According to Boryczka (2009), several modifications were introduced to improve the quality of clusters and, in particular, the spatial separation between clusters in the grid.

Changes that improve the spatial separation of clusters and allow a more robust algorithm were introduced by Handl, Knowles and Dorigo (2006). One is the restriction on function *f(i)* given by Equation (5), below, which serves to penalize high dissimilarities.

$$f^*(i) = \begin{cases} \dfrac{1}{\sigma^2} \sum\limits_{j \in L} \left[1 - \dfrac{d(i,j)}{\alpha}\right] & , \quad if \quad \forall j \quad \left(1 - \dfrac{d(i,j)}{\alpha}\right) > 0 \\ 0 & , \quad else \end{cases} \tag{5}$$

According to Vizine et al. (2005), a difficulty in applying the Clustering by Ants Algorithm on complex problems is that in most cases, they generate a number of clusters that is much larger than the real one. Moreover, these algorithms usually do not stabilize in a cluster solution, this is, they constantly construct and deconstruct clusters during the process. To overcome these difficulties and improve the quality of results, the authors proposed an Adaptive Ant Clustering Algorithm - A^{2}CA. An amendment included in this approach is a cooling program for the parameter that controls the probability of ants picking objects on the grid.

The spatial separation of clusters in the grid is crucial for individual clusters to be well defined, thus allowing for automatic retrieval. The spatial proximity, when it occurs, may indicate the early formation of the cluster Handl, Knowles and Dorigo (2006).

Defining the parameters of the neighborhood function is a decisive factor in the quality of clusters. In the case of the perception radius σ, it is more attractive to employ larger neighborhoods to improve the quality of the clustering and the distribution on the grid. However, this procedure is computationally more expensive (because the number of cells to be considered for each action grows quadratically with the radius) and also, it inhibits the rapid formation of clusters during the initial distribution phase. A perception radius that gradually increases with time accelerates the dissolution of preliminary small clusters Handl, Knowles and Dorigo, 2006. A progressive perception radius was also used by Vizine et al., 2005.

Moreover, after the initial clustering phase, Handl, Knowles and Dorigo (2006), replaced the scaling parameter $\frac{1}{\sigma^2}$ by $\frac{1}{N_{occ}}$ in equation (5), where N_{occ} is the number of grid cells occupied, observed within the local neighborhood. Thus, only the similarity and not the density was taken into account. In Boryczka (2009), the ACAM (Ant-based Clustering Algorithm Modified), proposed to replace the scalar $\frac{1}{\sigma^2}$ in equation (5) by scalar $\frac{\sigma_0^2}{\sigma^2}$, where σ_0 is the initial perception radius.

According to Handl, Knowles and Dorigo (2006), α determines the percentage of the grid patterns classified as similar. The choice of a very small value for α prevents the formation of clusters in the grid. On the other hand, the choice of a very large value for α results in the fusion of clusters.

Determine parameter α is not simple and the choice is highly dependent on the structure of the data set. An inadequate value is reflected by an excessive or extremely low activity in the grid. The amount of activity is reflected by the frequency of successful operations of an ant in picking and dropping. Based on these analyses, Handl Knowles and Dorigo (2006) proposed an automatic adjustment of α, as Boryczka (2009), who also proposed a new scheme for adjusting the value of α.

In Tan, Ting and Teng (2007), the authors examine the dissimilarity scalar parameter in Ant Colonies approaches for data clustering. The authors show that there is no need to use an automatic adjustment for α and they propose a method to calculate a fixed α for each database, calculated independently of the clustering process.

To measure the similarity between patterns, different metrics are used. In Handl Knowles and Dorigo (2006), the Euclidean distance is used for synthetic data and cosine for real data. In Boryczka (2009), different dissimilarity measures were used: Euclidean, cosine and Gower measures.

This paper is organized as follows: section 2 presents the basic ant-based clustering algorithm as proposed by Deneubourg et al. (1991) and the Ward method (JOHNSON; WICHERN, 1998), used for cluster recovery. Section 3 presents the method that was proposed (ACAP), with the major contributions (modifications and improvements) to Clustering based on Ant Colonies and, also, the assessment measures that were used on the clusters. Section 4 presents the two databases used to compare the two algorithms (ACAM and ACAP). Section 5 presents the results, discussions and illustrative figures of the ACAP and their performance compared to the results of applying the ACAM (BORYCZKA, 2009). Section 6 presents the final considerations.

2 THE BASIC ALGORITHM PROPOSED BY DENEUBOURG ET AL. (1991)

Initially, all patterns are randomly scattered throughout the grid. Then, each ant randomly selects a pattern to pick and is placed at a random position on the grid.

In the next phase, called the distribution phase, each ant is randomly selected. This ant travels the grid running one step of length L, in a direction defined at random. According to Handl, Knowles and Dorigo (2006), using a large step size speeds up the clustering process. The ant then probabilistically decides if it should drop its pattern at this position.

If the decision to drop the pattern is negative, another ant is randomly selected and the process is resumed. In case of a positive decision, the ant drops the pattern at its current position on the grid, if it is free. If this grid cell is occupied by another pattern, it must be dropped at a cell immediately adjacent thereto, which must be free, through a random search.

The ant then seeks for a new pattern to pick. Among the free patterns on the grid, this is, among the patterns that are not being carried by any ant, the ant randomly selects one, goes to its position on the grid, evaluates the neighborhood function and decides probabilistically whether it should pick this pattern. This process of choosing a free pattern on the grid runs until the ant finds a pattern that should be picked.

Only then this phase is resumed by selecting another ant until a stopping criterion is satisfied.

2.1 Cluster Recovery

The process begins with each pattern forming a cluster. After calculating the distances between all clusters the two clusters with the smaller distance should be blended (connected). According to Johnson and Wichern (1998), the most common types of connections are: Simple Connection, Complete Connection, Average Connection and the Ward Method (JOHNSON; WICHERN, 1998). The distances between clusters are defined in terms of their respective distances on the grid. Each pattern is now composed of only two attributes that place it on the two-dimensional grid. The distance between any two patterns is then the Euclidean distance between two points on the grid. This process repeats until a stopping criterion is satisfied.

When patterns around the edges of clusters are isolated, in Handl, Knowles and Dorigo (2006), was introduced a weight that encourages the fusion of these patterns with the clusters. The Ward Method used in this study makes the junction of two clusters based on "information loss". As the criterion for the "information loss" the square quadratic error (SQE) is considered.

3 PROPOSED CHANGES TO ANT-BASED CLUSTERING

In this section, we present the Ant-based Clustering Algorithm Proposed (ACAP), which are briefly discussed. This algorithm was based on the basic algorithm by Deneubourg et al. (1991) presented in section 2.

During the study of Ant-based Clustering, it was observed that many of the position moves on the grid patterns occur unnecessarily. It is considered an unnecessary move when a pattern is between similar ones and, in this case, there is no need to move this pattern to another position. In order to avoid these unnecessary moves we introduced a comparison between the probability of dropping a pattern at a position that was randomly chosen and the probability of dropping this pattern at its current position. The decision of dropping a pattern at the position that was randomly chosen only happens if this probability is higher than the probability of dropping this pattern at its current position.

We also noted the fusion of clusters that are close to one another on grid. When the decision to drop a pattern is positive and the cell in which the pattern should be dropped is occupied, a free random position close to this one is sought. However, this new position may also be close to another pattern cluster on the grid. This may be one reason for the merger of nearby clusters. Therefore, as an alternative to avoid the fusion of nearby clusters on the grid, we propose in this paper an assessment of the probability of dropping the pattern at the new position. The pattern is only dropped at the neighboring cell if the probability of dropping the pattern at this position is greater than the probability of dropping this pattern at its current position. All neighboring free positions are evaluated. If at no neighboring free position the probability of dropping the pattern is higher than the probability of dropping this pattern at its current location, the pattern is not dropped and the process resumes by choosing another ant.

Another issue observed in the Ant-based Clustering is that an ant can pick a pattern that is among similar ones on the grid. An ant only picks a pattern when it is not among similar ones on the grid, but from the moment the ant picks a pattern until the moment it is drawn to try and drop the pattern, changes may occur in its neighborhood and then may leave it among similar ones. Therefore, this ant is inactive because the operation to drop the pattern is not executed. In this case, it was proposed to replace the pattern picked by an ant if this pattern is not dropped within 100 consecutive iterations. The new pattern is chosen by lot, but it is only picked by the ant if the probability of picking this pattern is greater than 0.13397, a value that is discussed in Villwock and Steiner (2009a) and showed below. If there is no pattern with a picking probability greater than 0.13397, the last drawn pattern is picked by the ant. This could also be a stopping criterion.

The value 0.13397 was defined by making the picking probability (p_{pick}) equal to the dropping probability (p_{drop}). Figure 1 shows the picking and dropping probabilities chart.

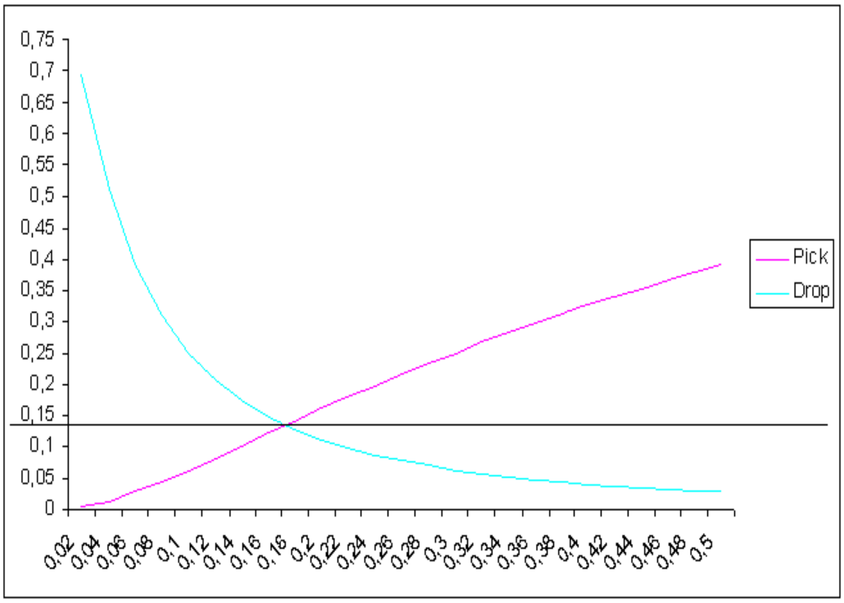

Figure 1 – Graphic of the probabilities in picking and dropping patterns
Source: The authors (2010)

When recovering clusters we used the Ward Method, once a maximum number of clusters was also defined. In Villwock and Steiner (2008), other methods were tested, among which the Ward Method yielded better results.

The ACAP was implemented with the computer software MATLAB (Matlab R2008). In this work we used LCPAD's computational grid resources: High Performance Central Laboratory/UFPR, which is partially financed by *FINEP* (*Financiadora de Estudos e Projetos*), project CT-INFRA/UFPR/Scientific Modeling and Computing.

3.1 Cluster Assessment

In assessing clusters, different aspects can be observed: determine the clustering trend of a data set, compare results from an analysis of clusters with results known externally, evaluate how well the results of a cluster analysis adjust to the data without reference to external information, compare results from two different sets of cluster analysis to determine which one is better, or even determining the correct number of clusters (TAN; STEINBACH; KUMAR, 2005).

According to Tan, Steinbach and Kumar, 2005, the numerical measures applied to judge several cluster evaluation aspects are classified into three types: external indexes are used to measure the extent to which cluster labels correspond to labels of classes supplied externally, internal indexes are used to measure how good the clustering structure is, unrelated to external information, and the relative indexes are used to compare two different clusters.

On her turn, Boryczka, 2009, used in her work two internal indexes (the Intra-Cluster Variance and Dunn's Index) and two external indexes (the *F* measure and the Random Index). These measures were also used in this paper.

3.2 ACAP PSEUDO-CODE

The following is the pseudo-code for the ACAP.

Initial phase

- Patterns are randomly scattered in the grid.
- Each ant randomly chooses a pattern to pick and is placed at a random position in the grid.

Distribution phase

- Each ant is selected randomly.
- This ant moves randomly along the grid.

- The ant probabilistically decides to drop its pattern in a position. The pattern is only dropped at the position randomly chosen if this probability is greater than the probability of dropping this pattern at its current position.
- If the decision is negative, another ant is selected at random and the distribution phase starts over again.
- If the decision is positive, the ant drops the pattern at its current position on the grid, if it is free.
- If this grid cell is occupied it must be dropped at a free neighboring cell through a random search. The evaluation of the probability of dropping the pattern at the new position is made and the pattern is only dropped at that neighboring cell if the probability of dropping the pattern at this position is still higher than the probability of dropping this pattern at its current position. If at no free neighboring position the probability of dropping the pattern is higher than the probability of dropping the pattern at its current location, the pattern is not dropped and the process starts again by choosing another ant.
- The ant randomly searches for a new pattern to pick (among the free patterns), goes to its position on the grid, evaluates the neighborhood function and decides probabilistically whether it shall pick this pattern.
- This choosing process of a free pattern in the grid runs until the ant finds a pattern that should be picked.
- The pattern an ant picks will be replaced if this pattern is not dropped after 100 consecutive iterations. Another pattern is randomly chosen, but the ant picks it only if the probability of picking this pattern is higher than 0.13397, a figure previously discussed. If there is no pattern with a picking probability higher than 0.13397, the ant picks the last pattern drawn.

Cluster recovery phase

- The process begins with each pattern forming a cluster.
- After calculating the distances between all clusters the two clusters with the shortest distance (these distances between clusters are defined in terms of their distance in the grid) should be merged (connected).

4 EXAMPLES TO EVALUATE THE ALGORITHMS' PERFORMANCES

In order to evaluate the performance of the Ant-based Clustering Algorithm Proposed (ACAP) compared to the Ant-based Clustering Algorithm Modified (ACAM), we used two databases, ÍRIS and WINE, whose data can be obtained in http://mlearn.ics.uci.edu/databases. The ÍRIS example consists of 150 patterns (plants).

In this example the clusters to which each plant belongs are known. The 150 patterns are divided into three clusters with 50 patterns in each cluster: Íris Setosa, Íris Versicolour and Íris Virginica. Each pattern consists of four numerical attributes: petal length, petal width, sepal length and sepal width.

The WINE example consists of 178 patterns (wines). In this example the clusters to which each pattern belongs are also known. The 178 patterns are divided into three clusters: 59 patterns belonging to cluster 1; 71 patterns belonging to cluster 2 and 48 patterns belonging to cluster 3. Each pattern consists of 13 numeric attributes, result of chemical analysis.

5 RESULTS AND ILLUSTRATION

The ACAP was applied to the two presented databases, ÍRIS and WINE (being known the cluster to which each pattern belongs, as described in section 4 above). As this is not an exact method, this is, there is variation in results when applied repeatedly, this method was applied to each database 10 times.

To evaluate the results the following measures were used to evaluate clustering: Random Index (*R*), *F* Measure and percentage of misclassification. Preliminary results for these databases were published in Villwock and Steiner (2009a) and Villwock and Steiner (2009b).

Table 1, below, shows the mean value and the standard deviation of the evaluating measures for the bases of real data that were analyzed. This table also presents the measures for evaluating the clustering for the best result among the 10 simulations.

Figure 2, below, shows the grid for the best result, whose evaluation measures are presented in Table 1 for the ÍRIS database. In this figure, the patterns represented by (#) belong to cluster 1, the patterns represented by (*) to cluster 2 and the patterns represented by (Δ) to cluster 3.

Table 1 – Results of applying the ACAP: the mean value of executing it 10 times, for bases of real data (ÍRIS and WINE) (VILLWOCK, 2009c)

Results		**R**	**F**	**Misclassification (%)**
ÍRIS	**Mean Value**	0,871	0,877	11,9

Results		R	F	Misclassification (%)
	Std Deviation	0,039	0,050	4,6
	Best Result	0,927	0,940	6,0
WINE	**Mean Value**	0,843	0,871	12,7
	Std Deviation	0,019	0,021	1,9
	Best Result	0,871	0,899	10,1

Source: The authors

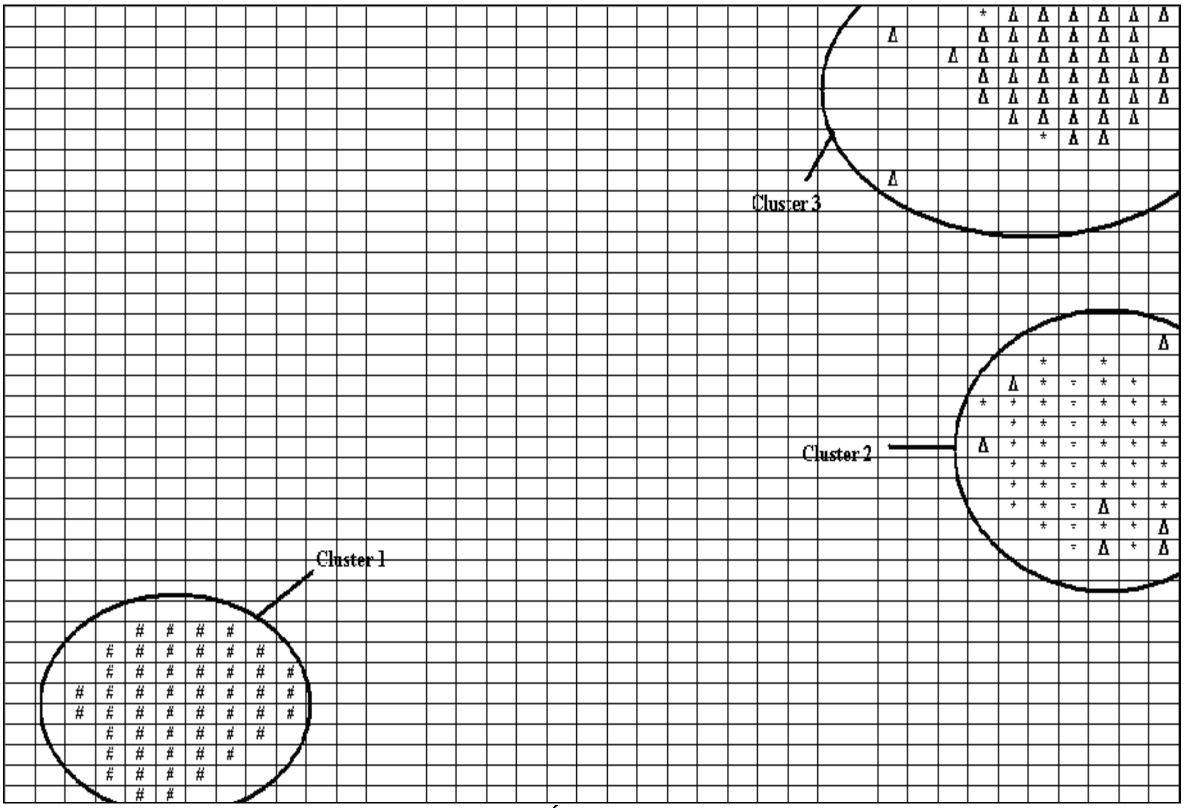

Figure 2 – Results of the ACAP for the ÍRIS database - best result (VILLWOCK, 2009c).
Source: The authors (2010)

Table 2 presents the comparison between the mean values of the measures for evaluating the clustering for the ACAP and the ACAM algorithms. The best results are in bold and show that the ACAP is better than the ACAM for the two databases analyzed.

Table 2 – Comparison between the average results of applying the ACAP and the results available in BORYCZKA (2009) (ACAM) for the databases

BASES	Assessment Measures	ACAM	ACAP
ÍRIS	**R**	0,819	**0,871**
	F	0,810	**0,877**
	Misclassification (%)	18,7	**11,9**
WINE	**R**	**0,849**	0,843
	F	0,868	**0,871**
	Misclassification (%)	13,9	**12,7**

Source: The authors (2010)

The results were satisfactory to the analyzed data bases. Analyzing the patterns distribution to the best results of the two data bases, it can be observed that, for the IRIS data base, only nine patterns, from 150, were misclassified. Besides that, as we can observe at figure 2, the group 1 contains all patterns assigned to it. For the WINE data base, 18 patterns, from 178, were misclassified.

6 FINAL CONSIDERATIONS

The collective and self-organizing behaviors of social insects inspired scientists to reproduce this behavior. The study of ant colonies has offered new ideas for clustering techniques. The Ant-based Clustering Algorithm has had special attention, once it still demands plenty more investigation to improve its performance.

This paper brings comments on three algorithms: 1. Ant-based Clustering Algorithm (original algorithm proposed by DENEUBOURG et al., 1991); 2. Ant-based Clustering Algorithm – Modified (ACAM, modifications proposed by BORYCZKA (2009) with respect to the original); 3. Ant-based Clustering Algorithm Proposed (ACAP, modifications proposed by the authors of this paper with respect to the original). The main modifications proposed here try to address known problems of the original algorithm, such as: too many unnecessary changes of patterns in the grid; incorrect merging of close clusters in the grid and change of a pattern that was already among similar ones in the grid.

The purpose with this paper was to evaluate the performance of modifications made to the Ant-based Clustering Algorithm (ACAP) compared to the ACAM (BORYCZKA, 2009).

In evaluating the performance of the algorithms, two real bases were used: ÍRIS and WINE. The results were very satisfactory to the analyzed data bases. Analyzing the patterns distribution to the IRIS and WINE data bases, it can be observed a low percentage of misclassified patterns (an average of 11.9% and 12.7%, respectively). In this way, the results showed that the ACAP proposed in this study was better than the ACAM, for both examples. This result, although quite satisfactory, still requires much investigation to improve its performance.

REFERENCES

BORYCZKA, U. Finding clusters in data: Cluster analysis with ants. **Applied Soft Computing**, v. 9, p. 61-70, 2009.

DENEUBOURG, J. L.; GOSS, S.; FRANKS, N.; SENDOVA-FRANKS, A.; DETRAIN, C.; CHRÉTIEN, L. The dynamics of collective sorting: Robot-like ants and ant-like robots. In:

Proceedings of the First International Conference on Simulation of Adaptive Behaviour: From Animals to Animats 1, p. 356-365, Cambridge, MA: MIT Press, 1991.

HANDL, J.; KNOWLES, J.; DORIGO, M. Ant-Based Clustering and Topographic Mapping. **Artificial Life**, v. 12, n. 1, p. 35-61, 2006.

JOHNSON, R.A.; WICHERN, D. W. **Applied Multivariate Statistical Analysis**. Fourth Edition. New Jersey: Prentice Hall, 1998.

MATLAB R2008 **–** The MatWorks, **MATLAB,** The MathWorks Inc., Natick, 2008.

TAN, P. N.; STEINBACH, M.; KUMAR, V. **Introduction to Data Mining**. Inc. Boston, MA, USA: Addison-Wesley Longman Publishing Co., 2005.

TAN, S. C.; TING, K. M.; TENG, S. W. Examining Dissimilarity Scaling in Ant Colony Approaches to Data Clustering. In: A**CAL**, 2007, Springer-Verlag, 2007.

VILLWOCK, R.; STEINER, M. T. A. Agrupamento Baseado em Colônia de Formigas: Estudo Comparativo de Algoritmos para Recuperação dos Grupos. XII Encontro Regional de Matemática Aplicada e Computacional, **Anais...,** Foz do Iguaçu, 2008.

VILLWOCK, R.; STEINER, M. T. A. Análise do Desempenho do Algoritmo de Agrupamento Baseado em Colônia de Formigas Modificado. XXXII Congresso Nacional de Matemática Aplicada e Computacional, Cuiabá, **Anais...**, SBMAC, 2009a.

VILLWOCK, R.; STEINER, M. T. A. Análise do Desempenho de um Algoritmo de Agrupamento Modificado Baseado em Colônia de Formigas. XLI Simpósio Brasileiro de Pesquisa Operacional - SOBRAPO, **Anais...,** Porto Seguro, 2009b.

VILLWOCK, R. **Técnicas de Agrupamento e de Hierarquização no Contexto de *KDD* –** Aplicação a Dados Temporais de Instrumentação Geotécnica-Estrutural da Usina Hidrelétrica de Itaipu. 125 f. Tese (Doutorado em Métodos Numéricos em Engenharia). Setor de Ciências Exatas, Universidade Federal do Paraná, Curitiba, Paraná, 2009c.

VIZINE, A. L.; DE CASTRO, L. N.; HRUSCHKA, E. R.; GUDWIN, R. R. Towards improving clustering ants: an adaptive ant clustering algorithm. **Informatica**, v. 29, p. 143-154, 2005.