Performance Analysis of an Ant-based Clustering Algorithm

Rosangela Villwock¹; Maria Teresinha Arns Steiner²; Pedro José Steiner Neto³

UFPR – Programa de Pós-Graduação em Métodos Numéricos em Engenharia  
²Coordenação do Curso de Engenharia de Produção; ³Departamento de Administração  
CP: 19081-CEP: 81531-990, Curitiba, PR, Brasil  
¹Universidade Estadual do Oeste do Paraná; Campus Francisco Beltrão; Rua Maringá, 1200, Vila Nova; CEP: 85.605-010; Francisco Beltrão; Paraná; Brasil

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 compared to the ACAM (Ant-based Clustering Algorithm – Modified version). 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 modified, two real examples were used: IRIS and WINE. The results show that the proposed algorithm (modifications) in this study was better than the ACAM for the two examples.

Keywords: Data Mining, Pattern 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 \( P_{\text{pick}} \) and \( P_{\text{drop}} \) given by equations 1 and 2 below, respectively.

\[
P_{\text{pick}} = \left( \frac{k_p}{k_p + f(i)} \right)^2
\]

\[
P_{\text{drop}} = \left( \frac{f(i)}{k_d + f(i)} \right)^2
\]

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, apud [2]), the authors used \( k_p = 0.1 \) and \( k_d = 0.3 \). In this paper, the authors obtained the estimate \( f(i) \) 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 [2]) 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 [11], 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 [2]), the decision of picking patterns is based on probability \( P_{\text{pick}} \) given by equation 1 above and the decision of dropping patterns is based on probability \( P_{\text{drop}} \) given by equation 3 below, where \( f(i) \) is given by equation 4.
In equation 4, \( d(i, j) \) is a function of dissimilarity between patterns \( i \) and \( j \) belonging to the interval \([0, 1]\), \( a \) is a scalar parameter dependent on the data (patterns) and belonging to the interval \([0, 1]\), \( L \) is the local neighborhood of size equal to \( s^2 \), where \( s \) is the perception radius (or neighborhood). In their work the authors used \( k_p = 0.1, k_d = 0.15 \) and \( a = 0.5 \).

Ant-based Clustering Algorithms are inspired mainly in the versions proposed by Deneubourg et al. (1991, *apud* [2]), and Lum er and Faieta (1994, *apud* [2]). Several modifications were introduced to improve the quality of clusters and, in particular, the spatial separation between clusters in the grid [1]. Changes that improve the spatial separation of clusters and allow a more robust algorithm were introduced by [2]. One is the restriction on function \( f(i) \) given by equation 5, below, which serves to penalize high dissimilarities.

\[
f^*(j) = \begin{cases} 
\frac{1}{\sigma^2} \sum_{j \in L} \left[ 1 - \frac{d(i, j)}{\alpha} \right], & \text{if } \forall j \left( 1 - \frac{d(i, j)}{\alpha} \right) > 0 \\
0, & \text{else}
\end{cases}
\]  

(5)

According to [11], 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 - A²CA. 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 [2].

Defining the parameters of the neighborhood function is a decisive factor in the quality of clusters. In the case of the perception radius \( s \), 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 [2]. A progressive perception radius was also used by [11].

Moreover, after the initial clustering phase, [2] replaced the scaling parameter \( \frac{1}{s^2} \) 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 [1], the ACAM (Ant-based Clustering Algorithm Modified), proposed to replace the scalar \( \frac{1}{s^2} \) in equation 5 by scalar \( \frac{s^2}{s^2} \), where \( s^2 \) is the initial perception radius.

According to [2], \( a \) determines the percentage of the grid patterns classified as similar. The choice of a very small value for \( a \) prevents the formation of clusters in the grid. On the other hand, the choice of a very large value for \( a \) results in the fusion of clusters.

Determine parameter \( a \) 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, [2] proposed an automatic adjustment of \( a \). On her turn [1] proposed a new scheme for adjusting the value of \( a \).

In [6], 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 \( a \). They propose a method to calculate a fixed \( a \) for each database. The value \( a \) is calculated independently of the clustering process.

To measure the similarity between patterns, different metrics are used. In [2], the Euclidean distance is used for synthetic data and cosine for real data. In [1] different dissimilarity measures were used: Euclidean, cosine and
Gower measures.

This paper is structured as follows: section 2 presents the basic ant-based clustering algorithm as proposed by Deneubourg et al. (1991, *apud* [2]) and the Ward method [3], used for cluster recovery. Section 3 introduces the databases that were used, implementation details of the method that was used, the major contributions (modifications and improvements) to Clustering based on Ant Colonies and the assessment measures that were used on the clusters. Section 4 presents the results, discussions and illustrative figures about the implementation of the proposed changes and their performance compared to the results of applying the ACAM [1]. Section 5 presents the final considerations.

2. The Basic Algorithm proposed by Deneubourg et al. (1991, *apud* [2])

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 [2], 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 [3], the most common types of connections are: Simple Connection, Complete Connection, Average Connection and the Ward Method [3]. 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 [2] 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. Methodology

The examples discussed were 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.

The proposed ant-based clustering analysis (modifications) was implemented with the computer software MATLAB [4]. In this work we used LCPAD’s computational grid resources: High Performance Central Laboratory/UFPR, which is partially financed by FINEP, project CT-INFRA/UFPR/Scientific Modeling and Computing. This algorithm was based on the basic algorithm by Deneubourg et al. (1991, *apud* [2]) presented in section 2. In this algorithm, some procedures have remained the same and several proposals for implementation were included in order to clarify it and improve its performance. These implementation details are described in [8]. When recovering clusters we used the Ward Method, once a maximum number of clusters was also defined. In [7], other methods were tested, among which the Ward Method yielded better results. Besides the implementation details described and the inclusion of improvements that have already been proposed previously, three main modifications were proposed and are described below.
3.1. Proposed Changes to Ant-based Clustering
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 [8]. 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.

3.2. 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 [5]. According to [5], 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, [1] 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 used in this paper.

4. Results
The proposed Ant-based Clustering Algorithm (modifications) was applied to two databases, IRIS and WINE (being known the cluster to which each pattern belongs, as described in section 3). 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 [8; 9].

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.

<table>
<thead>
<tr>
<th>Results</th>
<th>$R$</th>
<th>$F$</th>
<th>Misclassification (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRIS</td>
<td>Mean Value</td>
<td>0.871</td>
<td>0.877</td>
</tr>
<tr>
<td></td>
<td>Std Deviation</td>
<td>0.039</td>
<td>0.050</td>
</tr>
<tr>
<td></td>
<td>Best Result</td>
<td>0.927</td>
<td>0.940</td>
</tr>
<tr>
<td>WINE</td>
<td>Mean Value</td>
<td>0.843</td>
<td>0.871</td>
</tr>
<tr>
<td></td>
<td>Std Deviation</td>
<td>0.019</td>
<td>0.021</td>
</tr>
<tr>
<td></td>
<td>Best Result</td>
<td>0.871</td>
<td>0.899</td>
</tr>
</tbody>
</table>
Figure 1. below, shows the grid for the best result, whose evaluation measures are presented in Table 1 for the IRIS database. In this figure, the patterns represented by (#) belong to cluster 1, the patterns represented by (*) to cluster 2 and the patterns represented by (?) to cluster 3.

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

5. 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. The purpose with this paper was to evaluate the performance of modifications made to the Ant-based Clustering Algorithm compared to the ACAM [1].

The major changes were: replace the pattern carried by an ant in case it is not dropped within 100 consecutive iterations, compare the probability of dropping a pattern at a randomly chosen position with the probability of dropping this pattern at its current position, evaluate the probability for a new position if the pattern is not dropped at the position that was drawn, but at a neighboring position.

In evaluating the performance of the algorithm two real bases were used: IRIS and WINE. The results showed that the algorithm (modifications) 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.

Acknowledgements
The first author would like to thank CAPES for the scholarship.
References