A DOCUMENT CLUSTERING METHOD BASED ON ANT ALGORITHMS

ŁUKASZ MACHNIK

Departament of Computer Science, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland
lmachnik@elka.pw.edu.pl

(Received 29 December 2006; revised manuscript received 22 January 2007)

Abstract: Ant Algorithms, particularly the Ant Colony Optimization (ACO) metaheuristic, are universal, flexible and scalable because they are based on multi-agent cooperation. The increased demand for effective methods of managing large collections of documents is a sufficient stimulus to place the research on new applications of ant-based systems in the area of text document processing. The author presents an implementation of such a technique in the area of document clustering. Details of the ACO document clustering method and results of experiments are presented.

Keywords: ant algorithms, ant systems, document clustering, document grouping

1. Ant-based clustering methods

Researchers in the field of computer science have sought inspiration in nature to devise new efficient and effective algorithms for many years now, of which ant-based clustering is an instance. It works through positive feedback and local information processing, drawing its inspiration from the clustering and sorting behavior observed in real-life ants. Ant-based clustering and sorting was first introduced by Deneubourg in 1990 [1]. In the Monte Carlo model presented by Deneubourg, simulation ants are modeled by simple agents moving randomly in their environment, a square grid with periodic boundary conditions. Data items scattered in environment can be picked up, transported and dropped by agents. The picking and dropping operations of each individual agent are biased by the following probabilities:

$$\begin{aligned} p_{pick}(i) &= \left(\frac{k^+}{k^+ + f(i)}\right)^2, \\ p_{drop}(i) &= \left(\frac{f(i)}{k^- + f(i)}\right)^2. \end{aligned} \tag{1}$$

$f(i)$ is an estimation of the fraction of data items in an ant's immediate environment that are similar to the data item the ant currently considers to pick up or drop. Parameters k^+ and k^- determine the influence of the neighborhood function, $f(i)$. Many aspects of Deneubourg's model have been analyzed by other researchers [2–4].

2. Ant Colony Optimization

One of the issues explored in depth by ethnologist was understanding how almost blind animals were able to find the shortest way from their nest to food. Comprehending how this was achieved by nature was the first step to implement this solution in algorithms. The main inspiration to create the Ant Colony Optimization (ACO) metaheuristic was the research and experiments of Goss and Deneubourg [5].

Ants (*Linepithaema humile*) are insects living in communities called colonies. The primary goal of individual ants is the survival of the colony. A single specimen is irrelevant, only a community of ants is capable of efficient cooperation. Ants' cooperative abilities are based on the work of many creatures, evaluating one solution as a colony of cooperative agents. Individuals do not communicate directly. Each ant creates its own solution that contributes to the whole colony's solution [6]. The ability to find the shortest way between a source of food and the ant-heel is a very important and interesting behavioral trait of an ant colony. Ants have been observed to use a pheromone to mark the route they have already gone through. When the first ant randomly chooses a route it leaves a specific amount of pheromone, which gradually evaporates. Other ants looking for the way will choose the route where they feel more pheromone with greater probability and deposit their own pheromone there. The process is autocatalic in that the more ants choose a specific way, the more attractive it becomes for others. Further details can be found in publications of Marco Dorigo, who greatly contributed to the research of ant systems; they are the largest source of information on ACO [7, 6].

3. An ACO-based clustering method

The analogy between finding the shortest way by ants and finding most alike documents (the shortest way between documents) and the ability to use agents who construct their individual solutions as elements of the general solution became a stimulus to initiate research on applying ant-based algorithms in the document clustering process [8].

3.1. Adapting ACO concepts to the document clustering task

For the purpose of devising an effective method of clustering documents, it is necessary to modify and adjust the concepts specific to real-life ants, so that they can be effectively used to solve the problems of text mining.

- A colony of co-operating individual specimens:
  Artificial ants build a solution by moving along the graph of a problem, from one document to another. During each iteration number m of ants construct a solution in number n of steps, using a probabilistic law of making a decision. In practice, when upon visiting a specific document, i, an ant chooses the next document, j, to move to, a pair (i, j) is added to the solution being constructed. This step is repeated until the ant builds a complete solution for the specific iteration. Considering that this version of the algorithm is serial, having found a solution in a specific iteration process of leaving an amount of pheromone associated with a pair of documents, the ant dies. Yet new ants appear in her place, whose goal is to find a solution in the following iteration, deposit

pheromone and die. The pattern is repeated until the best result has been obtained or a specific amount of iterations performed.

- A pheromone trace and its influence:
  From the available variants of leaving pheromone on the path, the author has chosen a partial variant. Ants leave pheromone in a specific amount which equals a quotient of a constant and the length of the found path. In addition, the pheromone decays all partial solutions have been constructed – the sum of distances between all visited documents. The communication pheromone path is modified when a solution of a problem has been found merely to show the experience gained by ants while solving the problem.
- Finding the shortest path:
  The co-ordinate describing the location of a specific document in space will be a vector representing the frequency of words occurring in the document. In order to describe the distance between documents a simple measure will be used in multidimensional space, *viz.* the cosine distance. Finding the shortest path will be represented by finding such a sequence of passages from one document to anther that the sum of the reverse of cosine distances between the following elements of the examined set would be smaller. Using the reverse of cosine distance is necessary as an increase in cosine distance is evidence of greater similarity of documents. When the value of cosine distance is 0, it is suggested to add the ε parameter, almost nil in value, to the value of the similarity function in order to avoid division by 0.
- Accidental movement of individual ants in the initial phase of path-finding:
  Maintaining of this condition is necessary because in the initial phase of the algorithm the ants cannot use the experience of their predecessors. The pheromone trace between individual documents equals a selected constant value. Such situation enforces fully accidental choice of documents in the initial phase of path-finding.
- Artificial ants live in an artificial, discreet world and can move only from one specific position to another, *viz.* between states of the discreet world:
  The set of states between which agents can move will be defined as a set of vectors representing individual documents. As we have assumed earlier, each document will be represented by a vector based on the frequency of appearance of specific words in the examined text.
- The amount of pheromone deposited by an artificial ant is connected with a quality function of the solution achieved so far:
  The amount of pheromone deposited by ants is proportional to the quality of the solution they find: the shorter the distance between the documents, the greater the amount of pheromone deposited between them. It is also necessary to remember about the pheromone's evaporation and to exclude the stagnation phenomenon, *i.e.* choosing the same route by all ants too early.
- Memory of past states:
  Artificial ants are equipped with memory of passed states, which is supposed to prevent multiple location of one ant in the same position. (Otherwise ants could fall into cycles, which may prevent them from finishing the construction of the solution).

3.2. Details of processing

The method of document clustering introduced here is based on an artificial ant system [9, 10]. An application of such solution will be used as a method of finding the shortest path between documents, which is the goal of the first (or trial) phase of the considered method. The second phase, division, is separating a group of like documents.

The aim of the trial phase is to find the shortest path connecting every document in the set using the ACO algorithm [7, 6]. This is equivalent to building a graph, whose nodes would constitute the set of analyzed documents. The probability of choosing next document j by ant k occupying document i is calculated with the following function:

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [s_{ij}]^\beta}{\sum\limits_{k \in Z_k} [\tau_{ik}(t)]^\alpha \cdot [s_{ik}]^\beta}. \tag{2}$$

In the above formula, Z_k represents a list of documents not visited by ant k, $\tau_{ij}(t)$ represents the amount of pheromone in the trail between documents i, j, α is the intensity of pheromone trail parameter, β is the visibility of documents parameter, while s_{ij} is the cosine distance between documents i and j. After ants have completed their peregrinations, the pheromone trail evaporates and a new amount of pheromone is deposited between every pair of documents. The amount of pheromone that deposited by ants depends on the quality of the constructed solution (length of the path). In practice, adding pheromone to the trail and its evaporation are implemented with the formula presented below, adapted to every pair of documents (i, j):

$$\tau_{ij}(t) \leftarrow (1-\rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t). \tag{3}$$

In the above formula, $\rho \in \langle 0,1 \rangle$ stands for the pheromone trail's decay coefficient, while $\Delta\tau_{ij}(t)$ is the increment of pheromone between documents (i, j). The dependence controlling the amount of pheromone deposited by ant k between pair of documents (i,j) is as follows:

$$\Delta\tau_{ij}^k(t) = \begin{cases} n/L_k(t) & \text{for } (i, j) \in T^k(t) \\ 0 & \text{for } (i, j) \notin T^k(t). \end{cases} \tag{4}$$

In the above formula, $T^k(t)$ means a set of document pairs that belong to a path constructed by ant k, $L_k(t)$ is the length of a path constructed by ant k, while n is the amount of all documents. Finding the shortest path connecting every document in the set will be equivalent to building a graph, the nodes of which would constitute the set of analyzed documents. Like documents will be neighboring nodes in the graph, considering that the rank of individual nodes will be equal to or less than 2, which means that in the final solution one of the documents will be connected to only two other similar documents – each document in the designed solution would appear only once. Obtaining such a solution will end the first phase, known as *preparation*.

The code below represents the trial phase.

```
Procedure sequence_preparation()
{
    reset_pheromone();
    inialize_ants(number_of_ants);
    for (number_of_ants)
```

```
    {
        reset_ant();
        build_solution();
        update_best_document_sequence();
    }
    distribute_pheromone();
}
```

```
Procedure build_solution()
{
    while (available_documents)
    {
        update_ant_memory(current_document);
        compute_transition_probablities(current_doc, ant_mem);
        choose_document();
        move_to_next_document();
    }
    record_document_sequence();
}
```

The result of the trial phase is shown in Figure 1.

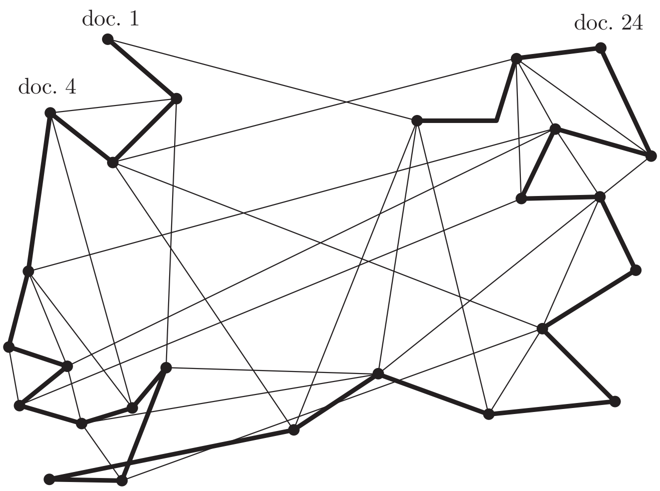

Figure 1. The influence of the attachment coefficient on the number of groups

In the following stage of the process it is necessary to separate a group of like documents from the sequence obtained in the first phase. The separation is achieved by appropriate processing of the sequence of documents (the shortest path) received in the preparatory phase. There are two ways to determine a document's assignment to a group.

3.2.1. Calculation based on the whole similarity measure

The vector representing the first document in the sequence is recognized as centroid μ of the first group to be separated. We calculate the cohesion of the first group using the whole similarity measure, $\|\mu\|^2$, in which the length of the centroid vector representing the considered group, square, is a measure of the group's cohesion. In the next step, the next document from the sequence is added to the first group.

The group's centroid μ and the change in its cohesion after adding the new element are calculated. If the change in cohesion has an acceptable value (less than $1-\delta$), the considered element becomes a permanent member of the group and we try to extend the group further by adding the next element from the sequence:

$$\frac{\left| \|\mu\|^2_{past} - \|\mu\|^2_{befor} \right|}{\|\mu\|^2_{befor}} < (1-\delta). \tag{5}$$

The δ parameter is called the attachment coefficient and its range is $(0, 1\rangle$. However, if the change in the group's cohesion following addition of the considered element has an unacceptable value (more than $1-\delta$), separating the first group is finished and the separation of the next (second) group begins. The vector of the document that could not be added to the first group becomes the initial centroid of the new group and the whole process is repeated from the beginning. Processing is complete when the whole sequence of documents has been dealt with.

3.2.2. Calculation based on distance

The vector representing the first document in the sequence is recognized as centroid μ of the first group to be separated. In the next step, we calculate the sum of all elements (positions) of the centroid vector, followed by the cosine distance between centroid vector μ and vector D representing the next element in the sequence of documents. Next, we check condition (6): if it is fulfilled, the considered element becomes a permanent member of the first group. We recalculate the value of the centroid and try to extend the group by adding the next element from the sequence:

$$\delta \cdot \sum_{k=1}^{n} t_{\mu k} < \cos(\mu, D). \tag{6}$$

The δ parameter is called the attachment coefficient and its range is $(0, 1\rangle$. However, if the condition is not met, the separation of the first group is finished and the separation of another (second) group begins. The vector of the document that could not be added to the first group becomes the initial centroid of the new group and the whole process is repeated from the beginning. Processing is complete when the whole sequence of documents has been dealt with.

The separation of groups is illustrated in Figure 2.

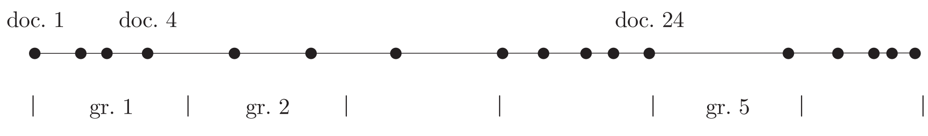

Figure 2. Separation of groups

The code below represents the division phase.

```
Procedure groups_separation()
{
    while (available_documents)
    {
        if (current_document==first_document)
        {
```

```
                new_group_creation();
                add_document_to_group(current_document);
                centroid_calculation(current_group);
            }
            else
            {
                if (check_attachment_condition)
                {
                    add_document_to_group(current_document);
                    centroid_calculation(current_group);
                }
                else
                {
                    new_group_creation();
                    add_document_to_group(current_document);
                    centroid_calculation(current_group);
                }
            }
        }
    }
```

3.3. Variants of the method

The amount of separated groups is strongly dependent on the attachment coefficient. When we use a large value of parameter δ (close to 1) we obtain numerous groups with a high degree of cohesion as a result of processing. A decrease in the value of δ yields a smaller amount of groups with less cohesion. In connection with above conclusion, there is a possibility to propose two variants of the considered method [11, 12].

One variant, which we shall refer to as single pass, is based on very precise execution of the trial phase: numerous ants. The duration of the first phase increases, but this enables acceptance of a smaller value of the attachment coefficient during the division phase and completion of processing after a single pass of the algorithm – a single trial phase and a single division phase.

The single pass variant of the clustering method is a non-hierarchical clustering method. Its main advantage is that the operator does not have to set the expected number of clusters at the beginning of processing. Results obtained in this variant are less precise than those of the other variant, but the processing time is much less. This variant of the method may also be used as the trial phase of other clustering algorithms, *e.g.* the separation of centroids for the K-means method.

The other variant, which we shall refer to as periodic, differs slightly from the variant discussed above. It assumes periodic processing of both trial and division phases. In every iteration of the division phase, a small number of neighbors are connected to form small groups. The value of the attachment coefficient is very high in the initial phases and gradually decreases to allow group creation in next iterations. During processing, each group is represented by a centroid. After groups have been formed and centroids calculated, the next iteration can be initiated, finding the shortest paths between the centroids and the documents. The whole process is complete when all documents are connected as a single cluster or when the stop criterion has been reached.

This variant is an agglomerative hierarchical clustering method that begins from a set of individual elements, which are then connected to the most similar elements forming ever larger clusters. The result of hierarchical processing is a nested sequence of partitions. The main partition is placed at the top of the hierarchy, including all elements from the considered collections. The base of the hierarchy is made of individual elements, while each middle level can be represented as a combination of clusters at a lower level of the hierarchy. The user can choose any level that will satisfy him as the solution.

3.4. Optimization

The second proposed variant is dynamic, which means that the optimal solution (the shortest path) is changed during each iteration. It is indicated to use an optimization method that adopts solution obtained so far to a new optimal solution of the problem. Its key aspect is the use of a solution obtained in previous phases (iterations) to find a solution to the changing problem. One of the dynamic problems solved using ant algorithms is the problem of finding a route in a telecommunication network [13, 14]. In the presented method (the periodic variant), a change (adding new calculated centroids) takes place at an exact point in time (next iteration) and the algorithm is required to adapt to the change. In the basic version of the presented method, the algorithm is reset once the problem has been changed (by adding new centroids and erasing documents grouped earlier). If we assume the change to be relatively small, it is probable that the new optimum will be connected with the old one. It can be useful to transfer knowledge obtained while creating the old solution to build the new one.

In order to realize the strategy described above, the author proposes to use modification of the pheromone trail between documents as a response to changing the problem: adding a new centroid and erasing the document. During modification of the pheromone trail, the problem is maintaining the right balance between resetting the correct amount of pheromone, to make the process of finding a new optimal solution flexible, and keeping enough knowledge to accelerate the search process.

Strategies of pheromone modification were presented inter alia in [15, 16]. Modifications described in these publications can be called global, but they have a disadvantage of ignoring where the change occurred. Therefore, the author proposes using the so-called η strategy, described in [17], to calculate the initial amount of pheromone trail for iterations $<> 1$. The η strategy uses heuristic information, *viz.* distance between documents, to define the degree of compensation that should be performed on a value of the pheromone trail. This method is based on implementing the function presented below to calculate the pheromone trail for every couple of documents/centroids (i, j):

$$\tau_{ij} \leftarrow (1-\gamma_i)\cdot\tau_{ij} + \gamma_i \cdot (n-1)^{-1}. \tag{7}$$

Parameter $\gamma_i \in \langle 1, 0\rangle$ is called the reset value and its value for every document/centroid is proportional to the distance between document/centroid i and a newly added element l. The value of the reset parameter is as follows:

$$\gamma_i = \max(0, d^s_{il}), \tag{8}$$

where

$$d_{il}^{s} = 1 - \left(\frac{s_{avg}}{\lambda * s_{il}}\right), \tag{9}$$

$$s_{avg} = [n*(n-1)]^{-1} \sum\nolimits_{p=1}^{n} \sum\nolimits_{k<>p} s_{kp}, \tag{10}$$

$$\lambda \in \langle 1, \infty \rangle. \tag{11}$$

Parameter n defines the number of elements taking part in the processing.

4. Experimental results

4.1. The experimental system

The main objective of our experiments was to confirm that ACO metaheuristic can be successfully implemented in processing text documents. In order to verify this thesis it was decided to compare the ACO clustering method with the three most popular and commonly implemented clustering methods. Additional tests were performed to compare ACO clustering with Ant-based clustering, as both of these methods are based on the behavior of real-life ants. The experiments presented here were executed using the KLASTERYZATOR_ACO document clustering system, implemented by ANSI C++. Two collections of documents were used in the research. One collection was *McCallum newsgroups* containing randomly chosen documents from twenty USENET forums. The other set was created by documents from the *Reuters-21 578* repository, representative of the largest thematic groups.

The most popular clustering methods were presented in [8]. Three of them were implemented in our experimental system: the K-means method (non-hierarchical), the single link method (hierarchical) and the average link method (hierarchical), chosen because of their popularity and common practical implementation. The Ant-based clustering algorithm was implemented as well to compare ACO clustering with Ant-based clustering.

The quality of results was evaluated using an internal quality measure of intra-cluster variance and purity, the latter being the ratio of the quantity of members of the dominating class and the size of the whole group. This method was chosen for two reasons. Firstly, the application of ant-based clustering in a real clustering task required evaluation of the obtained results without knowledge of the correct solution. Secondly, these functions provided additional information about the structure of the obtained solutions and were thus helpful in understanding and analyzing the results. Additionally, it is important to remember that the amount of groups obtained in processing was also a cluster evaluation measure. The presented method has unique properties of controlling the trend of the cluster creation number.

4.2. Number of groups

The ACO clustering method is characterized by an ability to identify the number of clusters in the processed collection. Most of the popular methods (K-means, single link, average link) require an input parameter constituting the number of outcome groups. This requires a priori knowledge of the collection to be processed or interaction with another algorithm, preparatory in function. Such interaction is very often a source of problems. At the same time, clustering algorithms able to identify the number of clusters automatically have many limitations. Incorrect choice of the number and value of centroids can have dramatic impact on the final solution, as can be observed in Figures 8–9.

At the same time, the impossibility of directly defining the number of resultant clusters can be seen as a disadvantage. There are many applications in which users require the possibility of defining this value themselves. Apart from identifying the number of resultant clusters, the clustering method presented here provides a tool to manipulate the trend of the cluster identification number, in the form of the attachment coefficient, δ. The flexibility in manipulating the number of clusters using the δ parameter is shown in Figure 3.

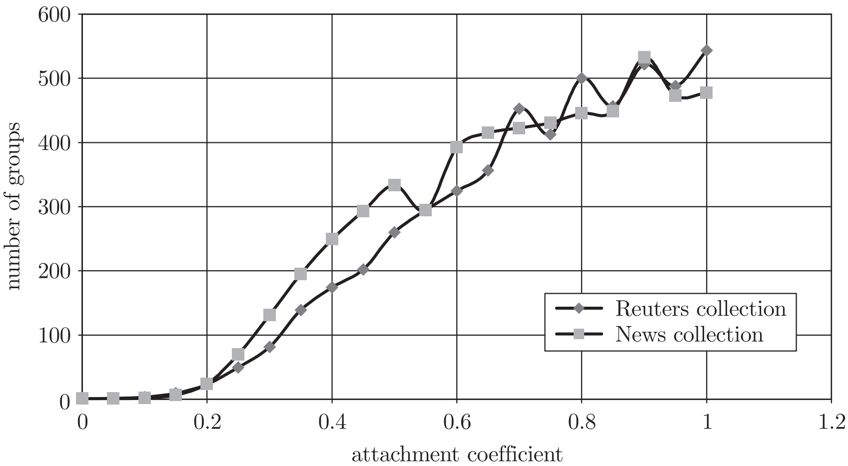

Figure 3. The influence of the attachment coefficient on the number of groups

4.3. Group sizes

The ways in which group sizes are obtained for the considered methods are shown in Figures 4–7. An analysis of the results suggests that the method proposed here is characterized by a proportional distribution of elements among clusters. A tendency to create one superior group is also noticeable. The results of ACO processing are quite similar to those of K-means processing. Importantly, the ACO clustering method has a tendency to limit the effect of creating a superior group instead of creating more balanced clusters with a high degree of cohesion (see Figures 8–9). The single link method and the average link method yield much poorer results then the other two methods, as they tend to create one predominant group. The Ant-based clustering algorithm has a disadvantageous tendency to create groups of the same sizes, no matter how documents are actually distributed among clusters.

4.4. Quality

The results' quality was evaluated using an internal quality measure of intra-cluster variance and purity, the latter being the ratio of the quantity of members of the dominating class and the size of the whole group. The experimental results presented in figures below demonstrate that the quality of ACO clustering is very high

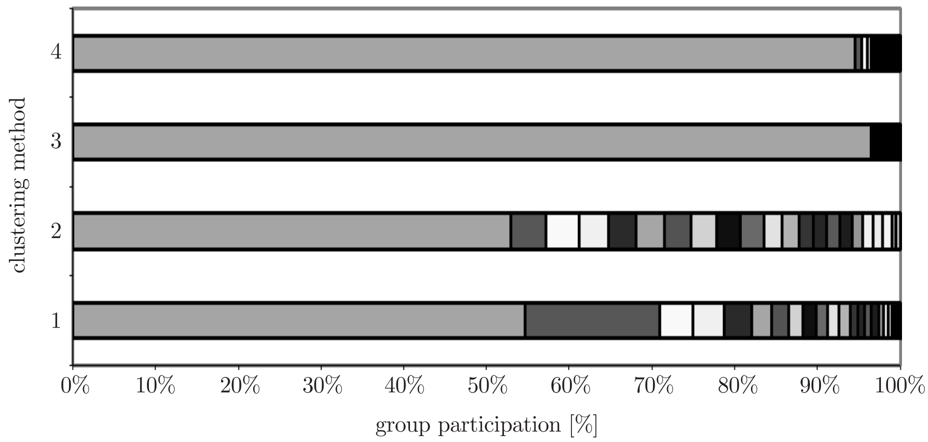

Figure 4. Group sizes for the news collection: (1) the ACO method, (2) the K-means method, (3) the single link method and (4) the average link method

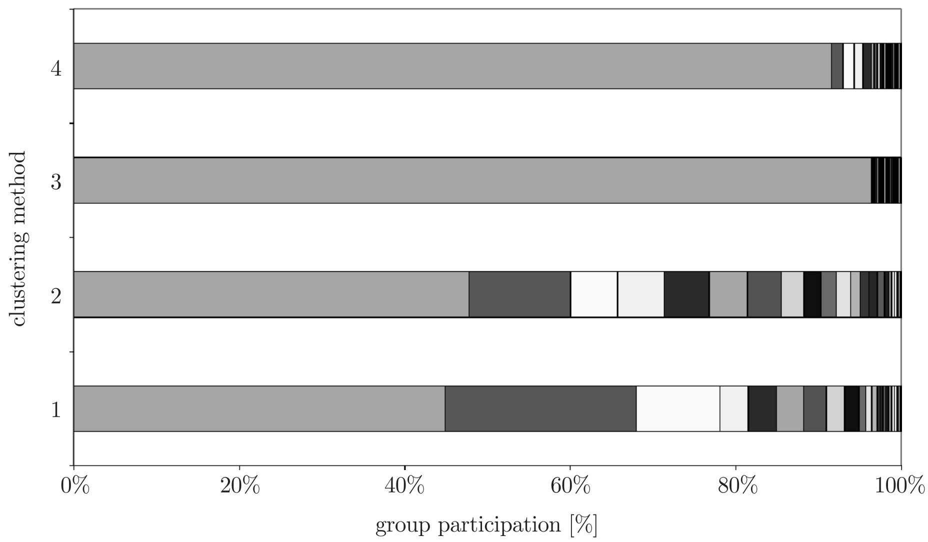

Figure 5. Group sizes for the Reuters collection: (1) the ACO method, (2) the K-means method, (3) the single link method and (4) the average link method

for both text collections. The results obtained for various numbers of groups demonstrate the dominance of the ACO method over the other tested methods. The ACO clustering's stable quality of results is noticeable.

The results generated by the single link and average link methods are quite similar, but there is a significant difference between them and the other results. We obtained good results of the K-means method for a small number of groups, but the quality of processing deteriorated at higher numbers, quite dramatically at the highest numbers. This effect is due to random selection of centroids and can be limited by using special algorithms for centroids generation. The results generated by the Ant-based method are slightly poorer than those of ACO clustering when variance and quality are considered.

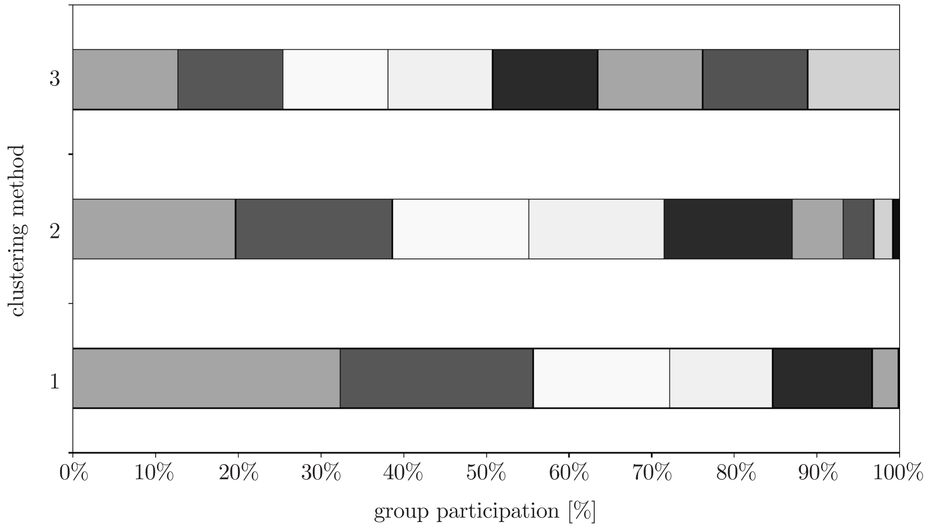

Figure 6. Group sizes for the news collection: (1) single pass ACO, (2) periodic ACO and (3) Ant-based

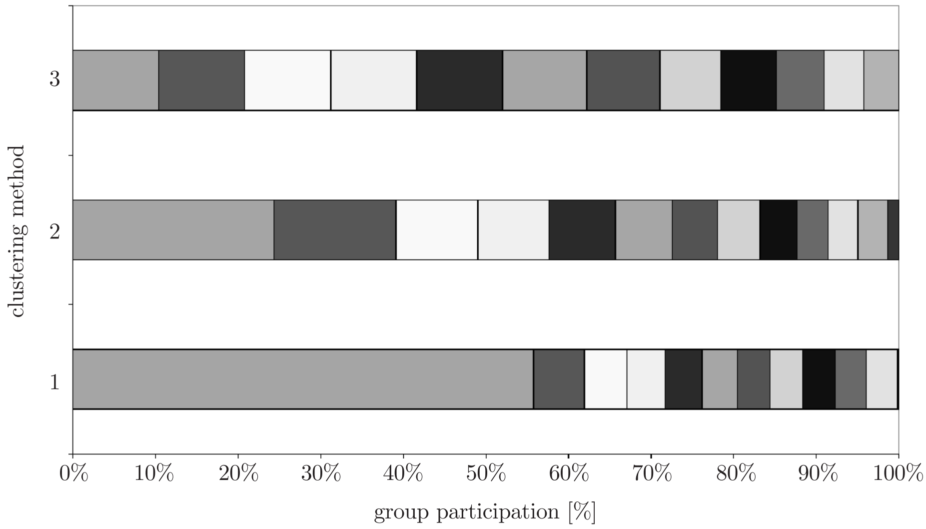

Figure 7. Group sizes for the Reuters collection: (1) single pass ACO, (2) periodic ACO and (3) Ant-based

4.5. Time

Our experiments have shown that the ACO method is much slower than the other tested methods for small collections of documents, while it is well ahead of its competitors for larger collections. Only the single link method is capable of producing its results faster than ACO, albeit with much poorer quality and group distribution.

The processing time of document collections of various sizes is shown in Figures 14–15. It depends on the number of resultant groups: the results of the best quality and good speed have been obtained only by the method here. Importantly, the fastest results have been generated by a quite small group of ants. This is connected

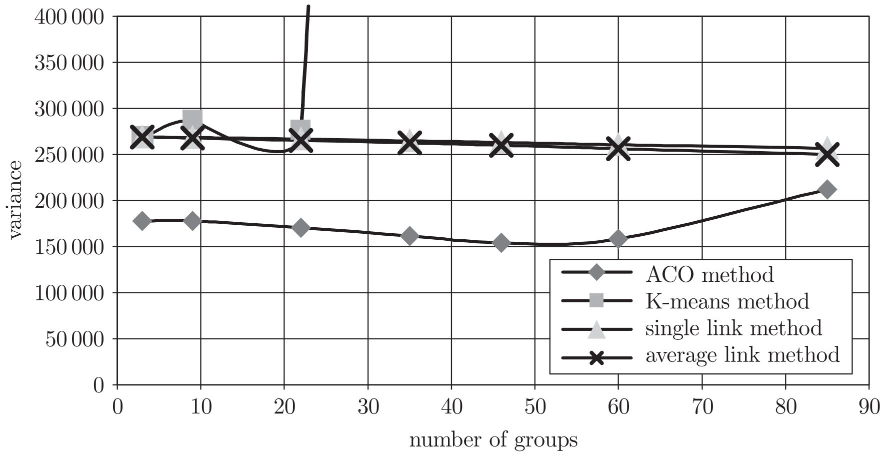

Figure 8. The value of variance: news collection

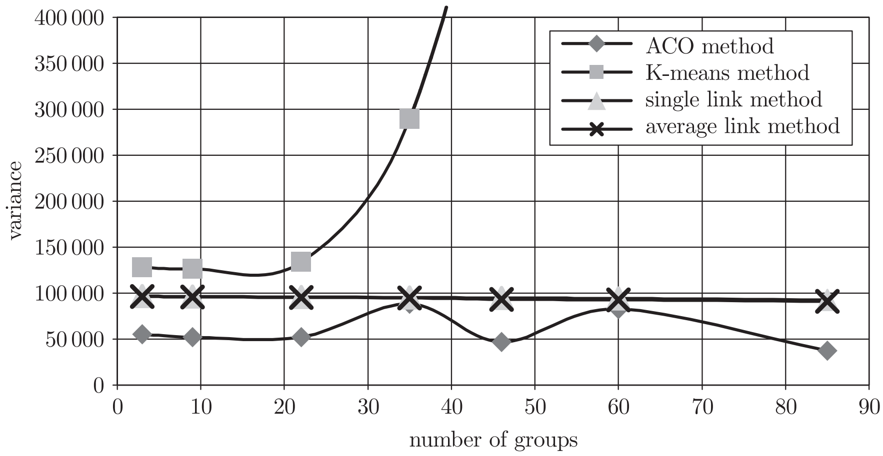

Figure 9. The value of variance: the Reuters collection

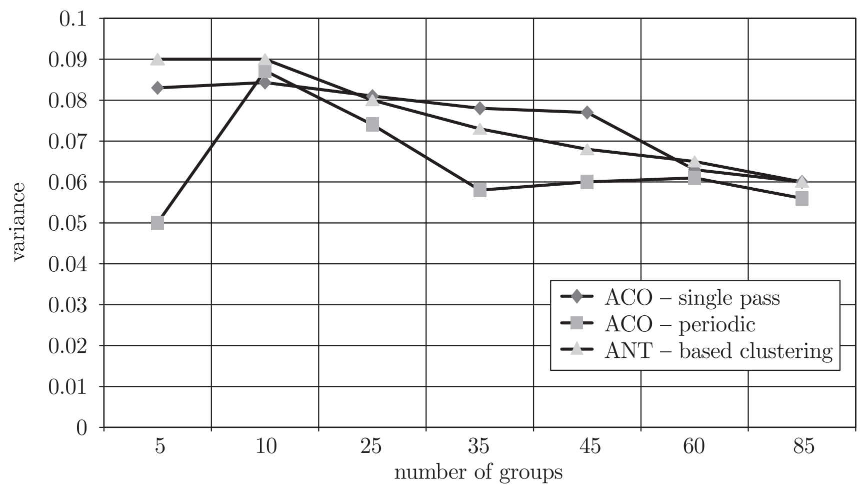

Figure 10. The value of variance: the news collection

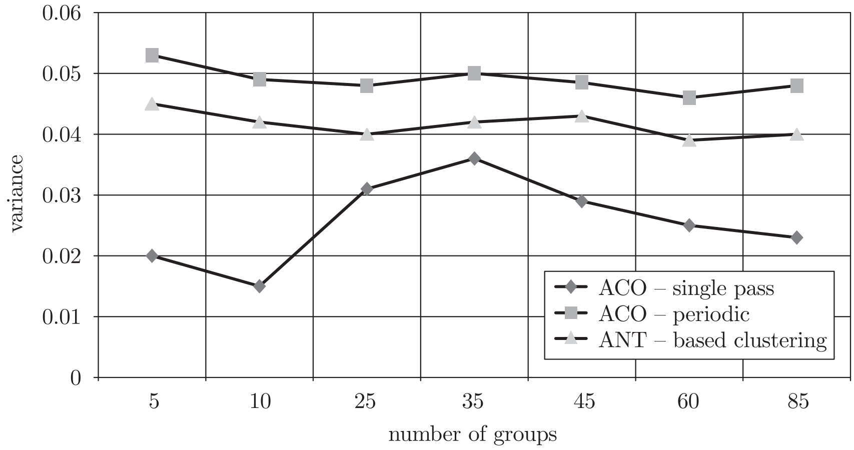

Figure 11. The value of variance:the Reuters collection

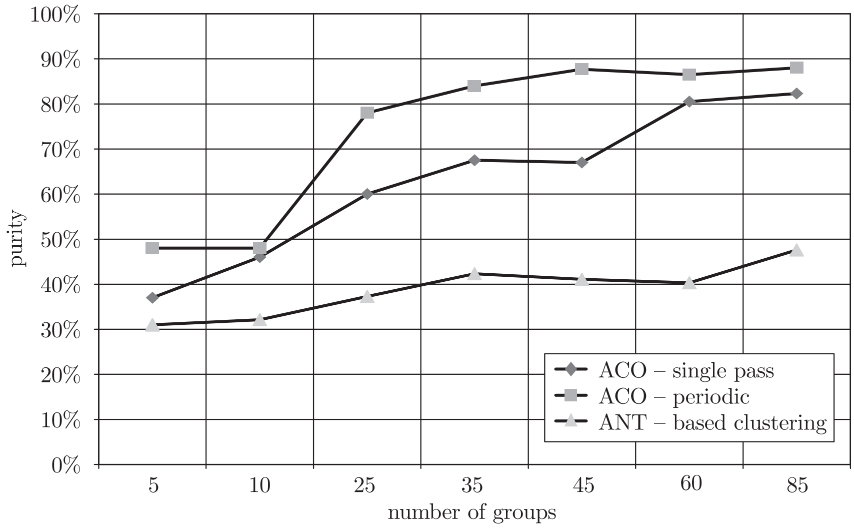

Figure 12. The value of purity: the news collection

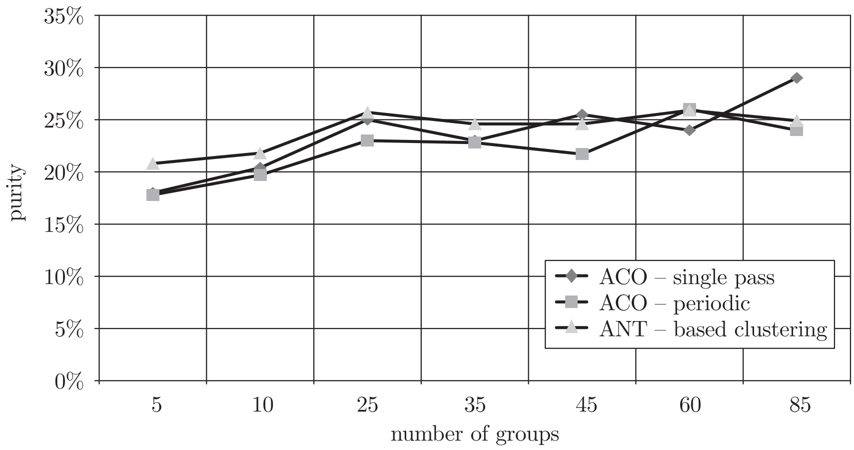

Figure 13. The value of purity: the Reuters collection

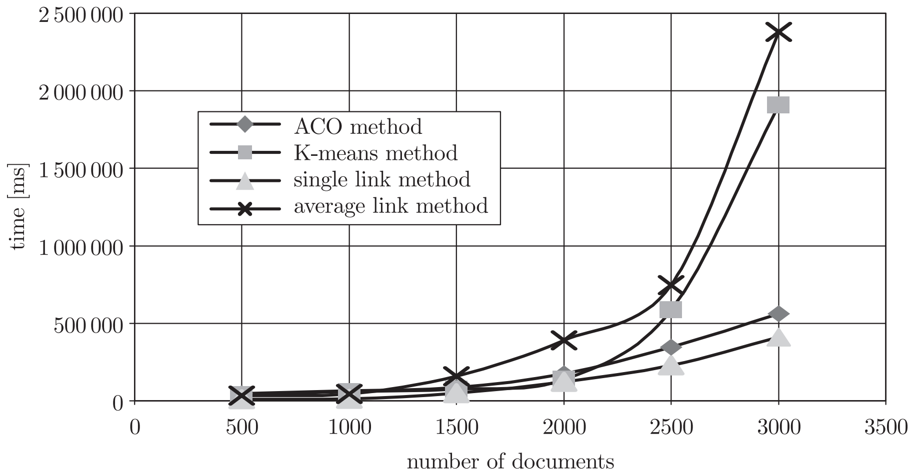

Figure 14. Relation between time and the number of processed documents

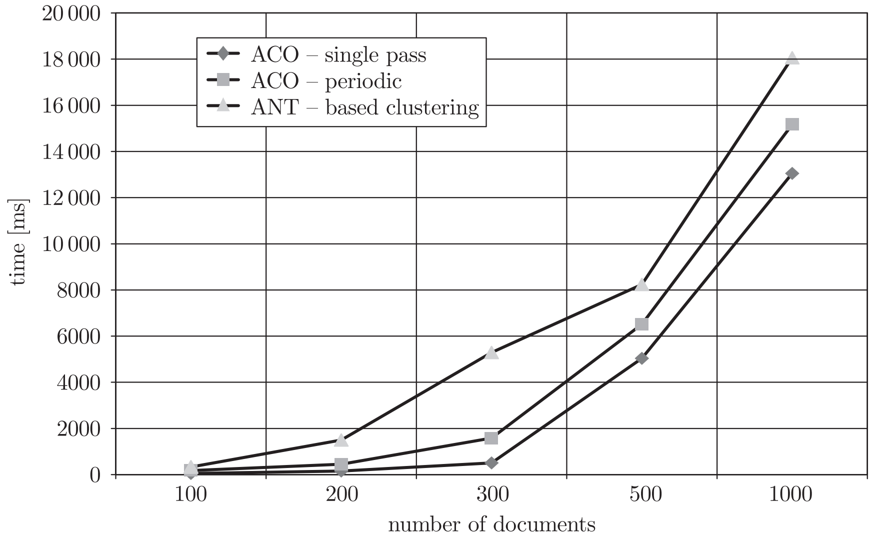

Figure 15. Relation between time and the number of processed documents

with a loss of quality, but even so the results are better than those obtained by the other methods.

5. Summary

Our experiments have confirmed that ant algorithms can be successfully implemented in processing text documents. A valuable clustering method based on the ACO metaheuristic has been devised, proving the universal nature and flexibility of this metaheuristic. Tests performed in a test environment have proved its usefulness and advantages. The obtained experimental results are characterized by good quality, speed for large collections of documents and flexibility in determining the number of resultant groups. There seems to be a possibility to increase the calculations' performance by implementing parallelization in processing, to be considered in forthcoming research. The main advantages of the ACO clustering algorithm are its abilities to determine the number of clusters in the collection and to manipulate the trend of the cluster identification number. Other aspects of using the ACO metaheuristic in document processing have been considered, incl. categorization using ACO, a method of identifying important documents in collections and a new way of querying collections.

References

[1] Deneubourg J-L, Goss S, Franks N, Sendova-Franks A, Detrain C and Chretien L 1991 *Proc. 1st Int. Conf. on Simulation of Adaptive Behaviour: From Animals to Animats 1*, MIT Press, MA, pp. 356–365

[2] Gutowitz H 1993 *3rd Europ. Conf. on Artificial Life*, MIT Press, Cambridge, MA, pp. 429–439

[3] Lumer E and Faieta B 1994 *3rd Int. Conf. on Simulation of Adaptive Behaviour: From Animals to Animats 3*, MIT Press, pp. 501–508

[4] Handl J 2003 *Ant-based Methods for Tasks of Clustering and Topographic Mapping: Improvements, Evaluation and Comparsion with Alternative Methods*, PhD Thesis, Friedrich-Alexander-Universität, Institut für Informatik

[5] Deneubourg J-L, Pasteels J M and Verhaeghe J C 1983 *J. Theor. Biol.* **105** 259

[6] Dorigo M, Maniezzo V and Colorni A 1996 *IEEE Trans. on Systems, Man, and Cybernetics – Part B* **26** (1) 1

[7] Dorigo M 1992 *Optimization, Learning and Natura Algorithms*, PhD Thesis, Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy (in Italian)

[8] Machnik Ł 2004 *Annales UMCS Informatica AI* **2** 401

[9] Deneubourg J-L, Goss S, Franks N, Sendova-Franks A, Detrain C and Chretien L 1991 *1st Int. Conf. on Simulation of Adaptive Behaviour: From Animals to Animats 1*, MIT Press, MA, pp. 356–365

[10] Lumer E and Faieta B 1994 *3rd Int. Conf. on Simulation of Adaptive Behaviour: From Animals to Animats 3*, MIT Press, pp. 501–508

[11] Machnik Ł 2006 *Advances in Systems, Computing Sciences and Software Engineering*, Springer, pp. 209–212

[12] Machnik Ł 2005 *Annales UMCS Informatica AI* **3** 315

[13] Di Caro G and Dorigo M 1998 *J. Artificial Intelligence Res.* **9** 317

[14] Schoonderwoerd R, Holland O, Bruten J and Rothkrantz L 1996 *Adaptive Behavior* **5** 169

[15] Gambardella L-M, Taillard E D and Dorigo M 1999 *J. Operational Research Society* **50** (2) 167

[16] Stützle T and Hoos H 1997 *Proc. Int. Conf. on Artificial Neural Networks and Genetic Algorithms*, Springer-Verlag, pp. 245–249

[17] Guntsch M and Middendorf M 2001 *Proc. EvoWorkshops, Lecture Notes in Computer Science*, Springer-Verlag, **2037**, pp. 213–222