Annales UMCS Informatica AI 5 (2006) 279-289

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI

http://www.annales.umcs.lublin.pl/

ACO documents clustering – details of processing and results of experiments

Łukasz Machnik*

Department of Computer Science, Warsaw University of Technology,
Nowowiejska 15/19, 00-665 Warszawa, Poland

Abstract

Ant algorithms, particularly Ant Colony Optimization met-heuristic, are a universal and flexible solution. In this publication the author presents the implementation of that technique in the documents clustering area – the new documents clustering method. The aim of this document is to present the details of the ACO documents clustering method, potential ways to optimize its processing and detail results of experiments.

1. ACO-based clustering method

Noticed analogy between finding the shortest way by ants and finding documents most alike (the shortest way between documents), and in addition ability to use agents who construct their individual solutions as an element of the general solution, became the stimulus to begin research on using the ant based algorithms in the documents clustering process [1].

1.1. Details of processing

The method of document clustering which is introduced here, is based on the artificial ant system [2,3]. Application of such a solution will be used as a method of finding the shortest path between the documents, which is the goal of the first phase (trial phase) of the method in question. The second phase (dividing phase) will have a task to actually separate a group of documents alike.

The aim of trial phase is to find the shortest path connecting every document in the set using ACO algorithm [4,5]. That is equivalent to building a graph, whose nods would make up a set of analyzed documents. The probability of choosing the next document j by ant k occupying document i is calculated by the following function (1).

* *E-mail address*: L.Machnik@ii.pw.edu.pl

$$p_{ij}^{k}(t) = \frac{\left[\tau_{ij}(t)\right]^{\alpha} * \left[s_{ij}\right]^{\beta}}{\sum_{k \in Z_k} \left[\tau_{ik}(t)\right]^{\alpha} * \left[s_{ik}\right]^{\beta}}. \tag{1}$$

In the above formula, Z_k represents a list of documents not visited by ant k, $\tau_{ij}(t)$ represents the amount of pheromone trail between documents i,j, α is the intensity of pheromone trail parameter, β is the visibility of documents parameter, however, s_{ij} is the cosine distance between documents i and j. After ants complete their trace the pheromone trail is evaporated and new amount of pheromone is left between every pair of documents. The amount of pheromone that is left by the ants is dependent on quality of the constructed solution (length of the path). In practice, adding the new portion of pheromone to trail and its evaporating is implemented by the formula presented below. This formula (2) is adapted to every pair of documents (i,j).

$$\tau_{ij}(t) \leftarrow (1-\rho) * \tau_{ij}(t) + \Delta\tau_{ij}(t). \tag{2}$$

In the above formula, $\rho \in (0,1>$ stands for the pheromone trail decay coefficient, while $\Delta\tau_{ij}(t)$ is an increment of pheromone between documents (i,j). Below the dependence (3) that controls the amount of pheromone left by ant k between the pair of documents (i,j) is presented.

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} n/L_k(t), & \text{for } (i,j) \in T^k(t) \\ 0, & \text{for } (i,j) \in T^k(t) \end{cases} \tag{3}$$

In the above formula, $T^k(t)$ means a set of document pairs that belong to the path constructed by ant k, $L_k(t)$ is the length of 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, whose nods would make up a set of analyzed documents. Documents alike would be neighboring nods in the graph, considering that the rank of the individual nods will fulfil the condition of being smaller or equal to 2, which means that in the final solution one of the documents would be connected to only two others (similar) – each document in the designed solution would appear only once. Gaining of such a solution would mean the end of the first phase, known as *preparing*.

The code below represents the trial phase.

```
1  Procedure sequence_preparation()
2  {
3     reset_pheromone();
4     inialize_ants(number_of_ants);
5        for(number_of_ants)
6        {
7           reset_ant();
```

```
8            build_solution();
9            update_best_document_sequence();
10        {
11    distribute_pheromone();
12 }
```

```
1  Procedure build_solution()
2  {
3     while (available_documents)
4        {
5        update_ant_memory(current_document);
6  compute_transition_probablities(current_doc,ant_mem);
7        choose_document();
8        move_to_next_document();
9        }
11    record_document_sequence();
10 }
```

In the following stage of the process it is necessary to separate a group of documents alike in a sequence obtained in the first phase. The separation of groups is obtained by appropriate processing of the sequence of documents (the shortest path) received in the preparing phase [6]. Following individual steps of that process is described. The vector that represents the first document in the sequence is recognized as centroid μ of the first group that is separated. In the next step we calculate the sum of all elements (positions) of the centroid vector. After that we calculate the cosine distance between the centroid vector μ and the vector D that represent the next element of documents sequence. Next, we check the condition (4). If it is true, then the considered element permanently becomes the member of the first group. We recalculate the value of centroid and try to extend this group by adding the next element from the sequence.

$$\delta * \sum_{k=1}^{n} t_{\mu k} < \cos\left(\mu, D\right). \tag{4}$$

The δ parameter is called the attachment coefficient and its range is (0,1>. However, if the condition is false, then the separation of the first group is finished and the separation of the next (second) group begins. The vector of the considered document that could not be added to the first group becomes the initial centroid of the new group. The whole process is repeated from the beginning. Processing is finished when the whole sequence of documents is done.

The code below represents the dividing phase.

```
1  Procedure groups_separation()
```

```
2  {
3  while (available_documents)
4     {
5        if (current_document==first_document)
6           {
7              new_group_creation();
8        add_document_to_group(current_document);
9        centroid_calculation(current_group);
10       }
11    else
12       {
13       if (check_attachment_condition)
14          {
15             add_document_to_group(current_document);
16             centroid_calculation(current_group);
17          }
18       else
19          {
20             new_group_creation();
21             add_document_to_group(current_document);
22             centroid_calculation(current_group);
23          }
24       }
25    }
26 }
```

1.2. Variants of the method

The number of separated groups depends precisely on the attachment coefficient. When we use a big value (close to 1) of δ parameter as a result of processing we receive a large number of groups with a high degree of cohesion. The decrease of δ value causes receiving smaller number of groups with less cohesion. In connection with the above conclusion, there is a possibility to propose two variants of considered method [6,7].

The first variant called by the author – single pass, is based on very precise execution of the trial phase – a lot of ants. The duration of the first phase increases, however, this activity permits to accept a smaller value of the attachment coefficient during the dividing phase and finishing processing after the single pass of algorithm – the single trial phase and the single dividing phase.

The clustering method that uses the single pass variant is the example of non-hierarchical clustering method. The main advantage of that method is that operator does not have to set the expected number of clusters at the beginning of processing. The results received in this variant are less precise than those from

the second variant, however, the time of processing is much shorter than that of the second proposed variant. This type of considered method can also act as a trial phase for other clustering algorithms. The example can be separations of centroids for K-means method.

The second variant called by the author – periodic, differs a little bit from that proposed earlier. It assumes periodic processing of both phases: trial and dividing. In every iteration of dividing phase the small numbers of neighbors are connected into small groups. The value of attachment coefficient is very high in initial phases and is gradually decreased to allow group creation in next iterations. Each group during processing is represented by centroid. After group creation and centroids calculations the next iteration can be started – finding the shortest path between centroids and documents. The whole process is finished when all documents are connected as a single cluster or when the stop criterion is reached.

This variant is an example of agglomerative hierarchical clustering method that begins with a set of individual elements which are then connected to the most similar elements forming bigger and bigger clusters. The result of hierarchical technique processing creates a nested sequence of partitions. The main partition is placed at the top of hierarchy. It includes all elements from the collections under consideration. The base of hierarchy creates individual elements. Every middle level can be represented as combination of clusters that are at the lower level in hierarchy. User can choose any level that satisfies him as solution.

1.3. Optimization

The second variant proposed by the author is the dynamic one. It means that during each iteration the optimal solution (the shortest path) is changed. The use of optimization method that adopts solution to changing optimum is recommended. The key aspect is to use solution that was received in the previous phases – the previous iterations; to find solutions to the changing problem. Till now one of the dynamic problems that was solved by using ant algorithms has been that of finding a route in the telecommunication network [8,9]. In the presented method (periodic variant), a change (adding new calculated centroids) takes place in the exact point of time (next iteration) and it is required that algorithm should adopt to the change. In the basic version of presented method after the problem is changed (adding new centroids and erasing the previously grouped documents) algorithm is reset. If we assume that the change of problem is relatively small, it is probable that the new optimum will be connected with the old one. It can be useful to transfer knowledge that was discovered during creating the old solution to build the new one.

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

The strategies of pheromone modification were presented inter alia in publications [10,11]. Modifications that were described in those publications can be called – global, but their disadvantage is the fact that they do not include place where the change occurred. According to that, to calculate the initial amount of pheromone trail for iterations <>1, the author proposes using the strategy that is called η-startegy, described in [12]. The „η-startegy" uses heuristic information, distance between documents, to define a degree of compensation that should be performed on a value of pheromone trail. This method is based on implementing the function that is presented below to calculate pheromone trail for every couple of documents/centroids (*i,j*):

$$\tau_{ij} \leftarrow \left(1-\gamma_i\right) * \tau_{ij} + \gamma_i * \left(n-1\right)^{-1}. \tag{5}$$

Parameter $\gamma_i \in <1,0>$ is called the reset value and for every document/centroid its value is proportional to the distance between the document/centroid *i* and the new added element *j*. The value of the reset parameter:

$$\gamma_i = \max\left(0, d_{ij}^s\right), \tag{6}$$

where

$$d_{ij}^s = 1 - \left(s_{avg} / \lambda * s_{ij}\right), \tag{7}$$

$$s_{avg} = \left[n * \left(n-1\right)\right]^{-1} \sum_{i=1}^{n} \sum_{k<>i} s_{ki} \tag{8}$$

$$\lambda \in \left\langle 1, \infty \right). \tag{9}$$

The parameter *n* defines the number of elements that take part in processing.

2. Results of the experiments

2.1. Experimental system

The experiments that are presented in this publication were perfprmed using the KLASTERYZATOR_ACO document clustering system. That system was implemented by ANSI C++. During the research two collections of documents were used. The first collection was *McCallum newsgroups* that contained documents from twenty forums from the USENET network. Documents were chosen randomly. The second set was created by the documents from the *Reuters-21578* repository. The documents from that collection were representatives of the biggest thematic groups.

2.2. Clustering algorithms

In [1] the most popular clustering method was presented. In the experimental system three of them were implemented: K-means (non-hierarchical), single link method (hierarchical) and average link method (hierarchical). These methods were chosen because they are popular and commonly implemented in practice and that is the reason why they were good candidates for comparison.

2.3. Results evaluation

The results of experiments were evaluated using the internal quality measure – intra-cluster variance. This method was chosen for two reasons. Firstly, the application of ant-based clustering in the real clustering task required the evaluation of the obtained results without knowledge of the correct solution. Secondly, these functions provided additional information about structure of the obtained solutions and can therefore help to understand and analyze results. Additionally it is important to remember that the number of groups that we received from processing was also a cluster evaluation measure. The method presented by the author has unique properties to control the trend of cluster creation number.

2.4. Number of groups

The ACO clustering method is characterized by the ability to identify the number of clusters in the collection that is processed. The majority of popular methods (K-means, single link method, average link method) require the input parameter that constitutes the number of outcome groups. This kind of behavior requires the ‘a priori’ knowledge of collection that will be processed or interaction with another algorithm that has the preparatory function. Such interaction is very often the source of many problems. Also, clustering algorithms that are able to identify the number of clusters automatically have many limits. Incorrect choice of number and value of the centroids can have dramatic impact on final solution. This kind of situation can observe in Figs. 4 and 5.

On the other hand, impossibility to directly define the number of resultant clusters can be recognized as a disadvantage. There are many applications in which the user requires the ability to define that value by himself. The clustering method presented in this publication beside identifying the number of resultant clusters, delivers a tool to manipulate the trend of cluster identification number. This tool is used for attachment of the coefficient δ. Fig. 1 describes flexibility in manipulating the number of clusters using δ parameter.

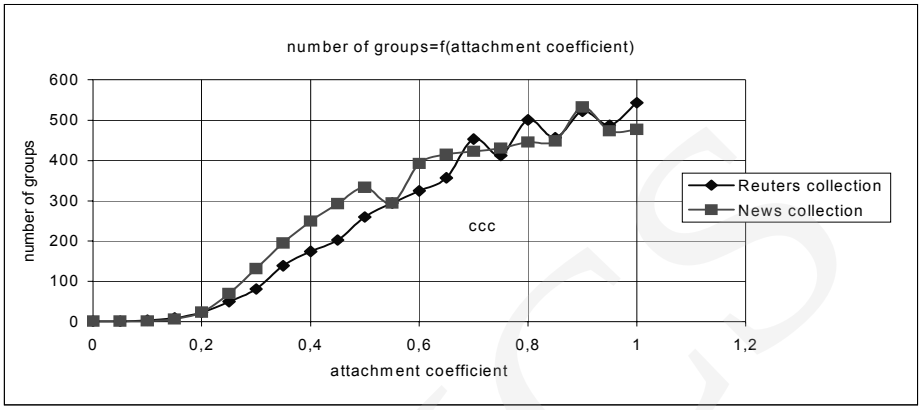

Fig.1. The influence of attachment coefficient on the number of groups

2.5. Sizes of the groups

Figs. 2 and 3 present the way of forming the sizes of the groups for the methods considered during experiments. The analysis of the results shows that the method proposed by the author is characterized by the proportional distribution of elements among clusters. Also, the trend of creating one superior group can be noticed. The results of ACO processing are quite similar to those for K-means processing. It is quite important to observe that the ACO clustering method has a tendency to limit the effect of creating one superior group instead of creating more balanced clusters with a high degree of cohesion (Figs. 4 and 5). The single link method and the average link method give much worse results then the first two methods. They have a tendency to create one predominant group.

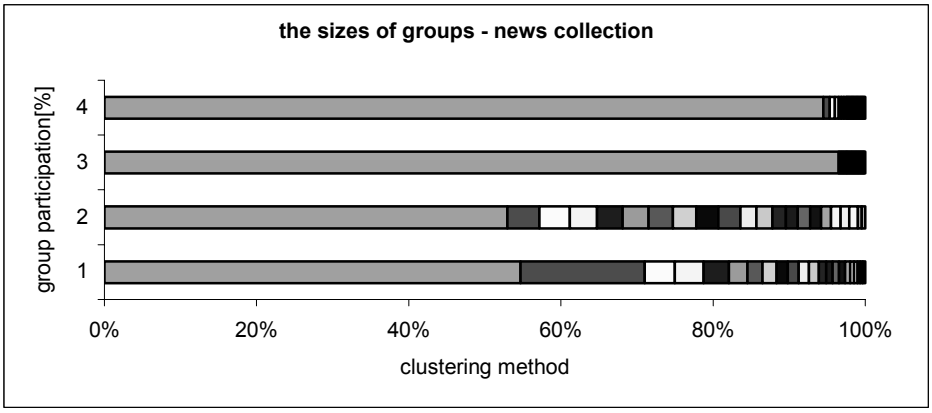

Fig. 2. The sizes of groups – news collection (1) ACO method, (2) K-means method, (3) single link method, (4) average link method

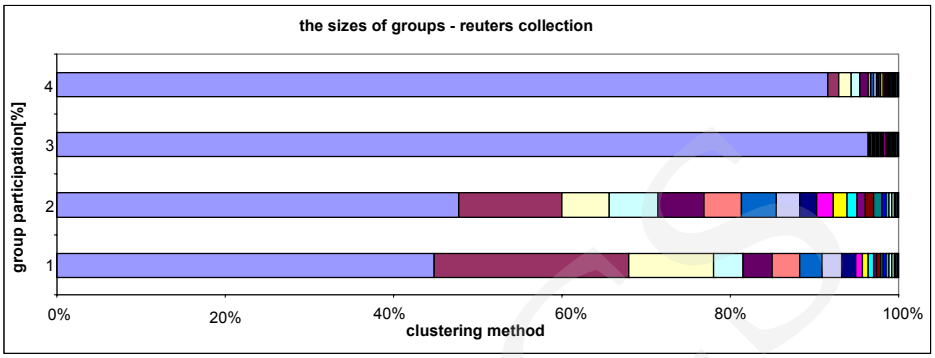

Fig. 3. The sizes of groups – reuters collection (1) ACO method, (2) K-means method, (3) single link method, (4) average link method

2.6. Quality

The results of experiments presented in the figures show that the quality of ACO clustering is very high for both texts collections. The results obtained for different numbers of groups demonstrate the dominance of ACO method over other tested methods. The quality stability of the results for ACO clustering should be noticed.

The results generated by the single link method and the average link method are quite similar but the difference between them and other results is significant.

For the K-means method we received good results for s small number of groups but the quality of processing is getting worse at a larger number of groups. For the K-means method we can also observe the dramatical deterioration of results quality with a very large number of groups. This effect is caused by random selection of centroids and can by limited be using special algorithms for centroids generation.

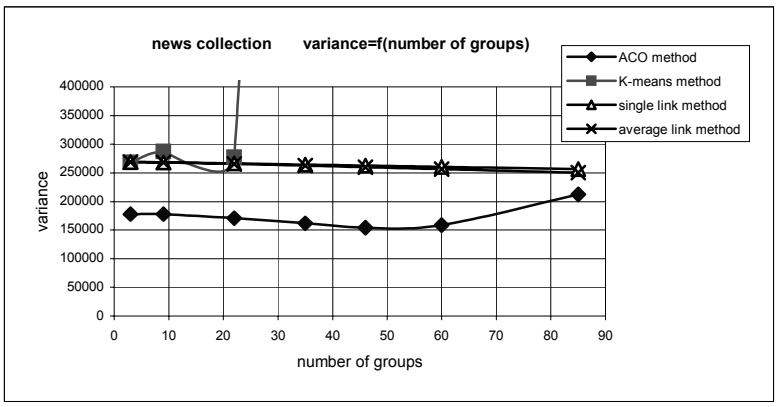

Fig. 4. The value of variance – news collection

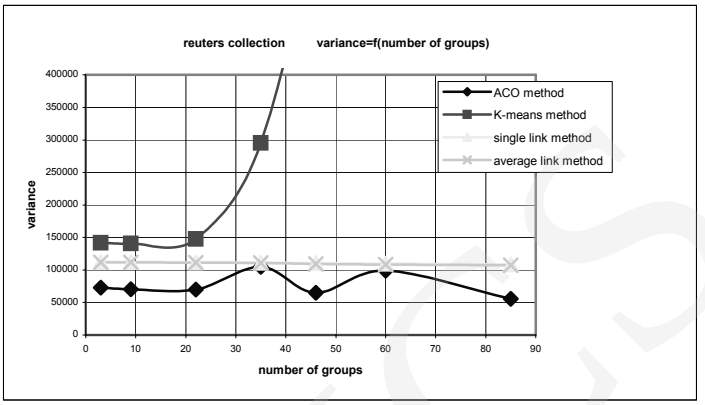

Fig. 5. The value of variance – reuters collection

2.7. Time

The experiments show that for small collections of documents the ACO method is much slower than the other tested methods. However, it should be noticed that the bigger size of collection the presented method tends to be ahead of the competitors. Only the single link method is able to return the results faster than the ACO method but at the same time the quality and group distribution is much worse.

Fig. 6 presents the time of processing for the documents collections with different sizes. The time of processing depends on the number of resultant groups. The results with the best quality and good speed are obtained only by the method proposed by the author. It is also important to note that the fastest results are generated using quite a small group of ants. It is associated with loss of quality but even so the results are still better than those obtained by other methods.

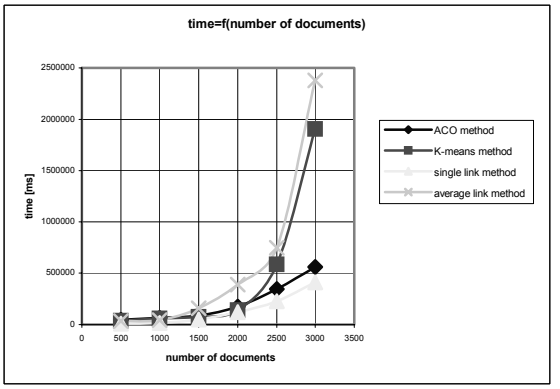

Fig. 6. Relation between time and number of processed documents

Conclusions

The experiments confirm an argument that the ant algorithms can be successfully implemented in the text documents processing. The attempt of creating valuable clustering method based on ACO meta-heuristic was successful. This proves the universal nature and flexibility of ACO meta-heuristic. The tests performed in test environment proved the utility and advantages of the method created by the author of this publication. The results obtained during experiments are characterized by good quality, speed for big collections of documents and flexibility in determining a number of resultants groups. It seems that it is possible to increase the performance of calculations by implementing a parallelization in processing. This topic will be dealt with in future research by the author.

References

[1] Machnik Ł., *Documents Clustering Techniques*, IBIZA 2004, Annales UMCS Informatica, (2004).
[2] Deneubourg J.-L., Goss S., Franks N., Sendova-Franks A., Detrain C., Chretien L., *The dynamics of collective sorting: Robot-like ants and ant-like robots*, First International Conference on Simulation of Adaptive Behaviour: From Animals to Animats MIT Press, MA, 1 (1991) 356.
[3] Lumer E., Faieta B., *Diversity and adaptation in populations of clustering ants*, Third International Conference on Simulation of Adaptive Behaviour: From Animals to Animats, MIT Press, 3 (1994) 501.
[4] Dorigo M., *Optimization, Learning and Natura Algorithms* (In Italia), PhD thesis Dipartimento di Elettronica e Informazione, Politecnico di Milano, IT, (1992).
[5] Dorigo M., Maniezzo V., Colorni A., *The ant systems: optimization by colony of cooperating agents*, IEEE Transactions on Systems, Man, and Cybernetics-PartB, (1996).
[6] Machnik Ł., *Ants in text documents clustering*, Proceedings of the International Conference on Systems, Computing Sciences and Software Engineering (SCSS 2005), (2005).
[7] Machnik Ł., *ACO – based document clustering method*, Konferencja Informatyka – Badania i Zastosowania, Kazimierz Dolny 2005, Annales UMCS Informatica, Poland, (2005).
[8] Di Caro G., Dorigo M., *AntNet: Distributed Stigmergetic Control for Communications Networks*, Journal of Artificial Intelligence, (1998).
[9] Schoonderwoerd R., Holland O., Bruten J., Rothkrantz L., *Ant-based Load Balancing in Telecommunications Networks*, Adaptive Behavior, (1996).
[10] Gambardella L.-M., Taillard E.D., Dorigo M., *Ant Colonies for the Quadratic Assignment Problem*, Journal of the Operational Research Society, (1999).
[11] Stützle T., Hoos H., *Improvements on the ant system: Introducing MAX(MIN) ant system*, Proc. of the International Conf. on Artificial Neural Networks and Genetic Algorithms, Springer-Verlag, (1997).
[12] Guntsch M., Middendorf M., *Pheromone Modification Strategies for Ant Algorithms applied to Dynamic TSP*, Proceedings of EvoWorkshops, Italy, (2001).