Improved Ant-based Clustering and Sorting in a Document Retrieval Interface

Julia Handl[1] and Bernd Meyer[2]

[1]FB Informatik, Universität Erlangen-Nürnberg
Julia.Handl@gmx.de
[2]School of Computer Science, Monash University, Australia
bernd.meyer@acm.org

Abstract. Sorting and clustering methods inspired by the behavior of real ants are among the earliest methods in ant-based meta-heuristics. We revisit these methods in the context of a concrete application and introduce some modifications that yield significant improvements in terms of both quality and efficiency. Firstly, we re-examine their capability to simultaneously perform a combination of clustering and multi-dimensional scaling. In contrast to the assumptions made in earlier literature, our results suggest that these algorithms perform scaling only to a very limited degree. We show how to improve on this by some modifications of the algorithm and a hybridization with a simple pre-processing phase. Secondly, we discuss how the time-complexity of these algorithms can be improved. The improved algorithms are used as the core mechanism in a visual document retrieval system for world-wide web searches.

1 Introduction

Ant-based sorting and clustering algorithms, introduced in a seminal paper by Deneubourg [Den90] and later extended by Lumer and Faieta [LF94], were among the first meta-heuristics to be inspired by the behavior of ants. Our interest in ant-based clustering is motivated from a concrete application perspective: we employ these methods as the core of a visual document retrieval system for world-wide web searches to classify online documents by contents-similarity.

The capability to perform a combination of clustering and multi-dimensional scaling [CC94], i.e. to generate a distance-preserving embedding, that has been ascribed to ant-based algorithms appears to make them particularly well-suited for our application. However, our experiments do not support the suggestion in the literature that "...a certain distance preserving embedding is guaranteed..." [KS99] (Here goal was to find a distance preserving embedding of graphs into a metric space). Instead the results show that the original algorithms perform effective clustering, but have only limited capability for multi-dimensional scaling. We demonstrate how this can be improved upon with some modifications of the algorithms and a simple pre-processing stage. We also demonstrate modifications of the algorithms that significantly improve their run-time, making them acceptable for interactive applications, such as online searches.

1.1 Contents-based Document Clustering: Topic Maps

Insufficiently specific world-wide web searches often return thousands of documents. To help the user orient in such large document collections, it has proven useful to classify documents according to contents-similarity and to visualize the

classified document collection in the form of a *topic map* [Fab00]. On a topic map semantically similar documents appear in spatial proximity, whereas unrelated documents are clearly separated. Documents are clustered around topics represented by mountains in the landscape. "Height" of a document corresponds to its relevance for the topic and topic labels serve as landmarks.

Our hybrid ant-based clustering method is the core mechanism of a fully implemented search-interface that operates as a front-end to the popular search engines Google and HotBot. Users specify a full text query which is passed on to the back-end search engine. Matching documents returned by the search engine are classified based on either a full text analysis or only on the contents of the snippets returned by the search engine and a topic map is generated to visualize the entire collection of documents (Fig. 1). Starting from these maps, the user can conveniently explore the query results by browsing the documents, which are accessed through a mouse click on their map location. Due to space limitations this paper focuses on the clustering algorithm and will not discuss details of the interface or the pre-processing.

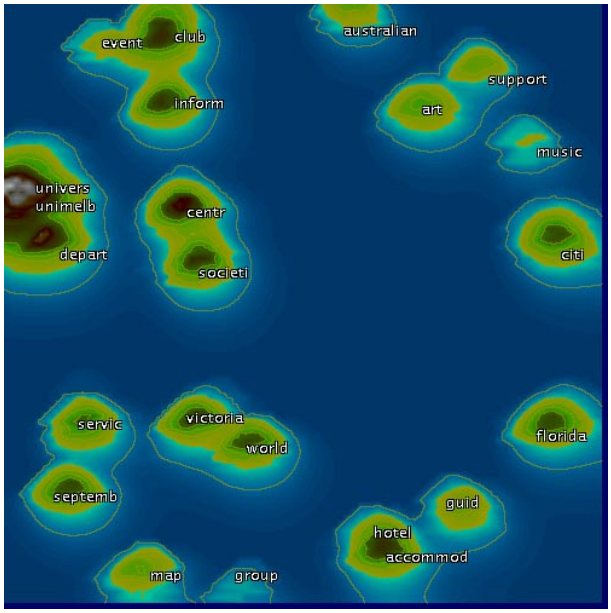


Fig. 1. Ant-generated Topic Map (1000 documents, Query "Melbourne", Search Engine "Google")

Essentially there are two central steps involved in generating topic maps: (1) The documents must be given positions in a high-dimensional conceptual document space that reflects their contents adequately; (2) The document space needs to be embedded into the 2-dimensional visualization space. Ideally, this embedding should combine multi-dimensional scaling with clustering. *Clustering* of the document data helps the user to identify the main structure and to focus on the main topics in the collection. *Multi-dimensional scaling* is necessary, as it provides meaningful inter-cluster relations. On an ideal topic map, clusters with similar contents should be in close proximity, while differences in subject should be reflected by spatial distance.

The idea of a map metaphor for document retrieval as such is not new and quite a few interfaces that use this visualization have been proposed (see Section 4). However, these systems have only dealt with comparatively small sets of documents or with precompiled (offline) data. Systems that have performed fast document classification online "on the fly", such as [CD00,ZE99], had to resort to simpler forms of document classification. To our knowledge, our system is the first to generate large topic maps dynamically for online-queries.

2 Clustering versus Multi-dimensional Scaling

Let us first make precise the embedding task that the application poses. A set of n relevant keyword terms is selected to classify the documents (see [Han] for details). Each document is positioned in the resulting n-dimensional document space according to its use of keywords. The k-th component of the characteristic vector for document i is computed using normalized IDF weighting [Sal88]:

$$d_{ik} = \frac{tf_{ik} \times log(N_D/n_k)}{\sqrt{\sum_{j=1}^{N}(tf_{ij} \times log(N_D/n_j))^2}}$$

where N_D is the size of the entire document collection; n_i is the number of documents containing the specific term i; and the term frequency tf_{ik} is $tf_{ik} = log(f_{ik})+1$ or $tf_{ik} = 0$ if $f_{ik} = 0$. Here f_{ik} is the frequency of word k in document i. To reduce the dimensionality of the document space we apply *Latent Semantic Indexing* [DDL+90] as a post-process. In the resulting document space, Euclidean distance is an adequate measure of the contents-similarity of two documents.

The second step in generating a topic map is to find a distance-preserving clustering of the high-dimensional document space into the 2-dimensional display space. Ant-based sorting-and-clustering seems a prime candidate to compute this kind of embedding. Let us briefly recall the original algorithm. The ants act on a two-dimensional torus grid on which documents are initially assigned random positions. Each grid cell can at any time only be occupied by a single document. Ants perform two probabilistic actions. (a) *Picking*: if an unloaded ant steps on a field occupied by a document, it may pick up this element; (b) *Dropping*: an ant carrying a document can, at any stage, drop this element on a free cell. At each iteration of the algorithm an ant performs a picking or dropping action or neither according to the probabilistic decision and subsequently steps on a randomly chosen immediately neighboring grid cell. The probability of picking or dropping is influenced by the ants local perception of its environment: $p_{drop}(i) = \left(\frac{f(i)}{k_d+f(i)}\right)^2$; $p_{pick}(i) = \left(\frac{k_p}{k_p+f(i)}\right)^2$ where k_d and k_p are adjustable parameters ($k_d = k_p = 0.1$ in our experiments) and

$$f(i) = max\left(0,\ \frac{1}{|S|}\sum_{\{c_{ij}\in S | d_{ij}\neq nil\}}\left(1 - \frac{d(k,d_{ij})}{\alpha\ \mu}\right)\right)$$

where d_{ij} is the index of the document in cell c_{ij} and S is the neighborhood around the ant's current position (5×5 in our experiments). $d(k,d_{ij})$ is the similarity (distance) between the document with index d_{ij} and the document k currently carried or considered by the ant. N is the number of documents in the collection and the scaling factor is $\mu = \frac{2}{N(N-1)}\sum_{k=1}^{N}\sum_{l=1}^{k-1}(d(k,l))$. The parameter $\alpha \in [0,1]$ permits further adjustment of the resulting values. It will later play a crucial role in our modifications of the algorithms.

As discussed above, the embedding to generate a proper topic map requires (a) adequate clustering and (b) that the distances in the visualization space strongly correlate with the distances in the document space. *Do ants really generate a distance preserving embedding?* In [LF94] the original algorithm was mainly evaluated using visual observation and by measuring spatial entropy. The quality of sorting was measured by the global fit and a dissimilarity measure. As these do not reflect the global correlation well enough, [KSL98] used the overall Pearson Correlation (the degree of linear relationship between two

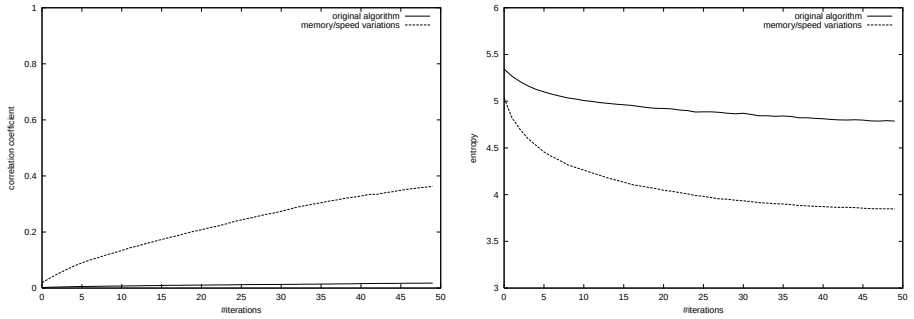


Fig. 2. Inter-cluster correlations (left) and Entropy (right) (mean values for 50 runs).

variables) as an additional performance measure. Here, correlations of up to 0.65 were reported but not analyzed further.

Importantly, it must be noted that the method is very sensitive to the choice of α and correlations like 0.65 are only achieved with the proper choice of α. This problem was already noted in [KSL98], but no method was suggested how the proper α value could be found without resorting to manual experimentation. As we will discuss later, the proper choice of α depends on the structure of the data and the solution is to introduce an adaptive strategy.

To further analyze the performance, we measure Pearson correlation on several levels: (1) Overall correlation of all elements; (2) Inter-cluster correlations, where the weighted average of all cluster elements is used as cluster center; (3) Intra-cluster correlations, which are observed by computing the correlations for the clusters individually. In our experiments we have additionally measured Spearman rank correlation, which is better suited to track non-linear relationships. However, as the experimental results generally confirm the observations made for the Pearson correlation, we do not report them in this paper.

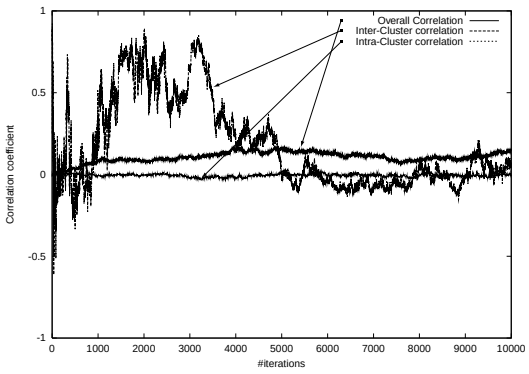


Fig. 3. Overall correlation and inter/intra cluster correlation (single run, 800 element set).

In the original studies in [LF94] good results were reported for an artificial test data set based on visual inspection. Initially, these findings were confirmed by the correlations measured in our experiments. However, more extensive experiments reveal that these results are mostly due to the simplicity of the test data.The test data is composed of only four normal distributed clusters (100 elements each) around the centers (0,0); (0,8); (8,0); (8,8) with an identical standard deviation of 2 and sorted on a grid of size of 52×52. Due to this limited grid size, the limited number of elements and the relatively good spatial separation of the clusters, the algorithm can recover the cluster structure too easily.

In further experiments with larger grid sizes and more complex test data the entropy is still reduced, but the data leave very serious doubts about the quality of the obtained embedding. The modified test data consists of four clusters of 200 elements each on a grid of size 100×100. Cluster centers are now irregularly distributed at (0,0); (0,80); (8,0) and (30,30).

Figure 2 shows the development of overall correlation and entropy on the modified test data set for the original algorithm and the algorithm with speed and memory improvements (see Section 3) averaged over 50 runs. It is clearly visible that the correlation remains very low. When analyzing the different correlation types further we can see that intra-cluster correlation and overall correlation both stay low. Even more interestingly, in individual runs the inter-cluster correlation is not stabilizing but oscillates (Fig. 3). This effect is also visible in a standard deviation of about 0.4 for the inter-cluster correlation data.

We conclude that the demands of multi-dimensional scaling for generating topic maps cannot sufficiently be met by the original algorithms, as the distances between individual clusters are not represented correctly. We also have to face the problem that a suitable fixed α-value cannot be determined in advance.

3 Algorithm Modifications

We now discuss our modifications to these algorithms that improve both performance and run-time. Lumer and Faieta already suggested two modifications to the algorithm to improve the quality of the results: (1) Each ant keeps a short-term memory of its most recent drop locations. This allows to bias the ant's movement after a pick-up in the direction of the "closest match" among the last n drop locations. In [LF94] it was suggested (through visual inspection of the results) that this modification has a significant influence on time and quality. As expected, this suggestion is confirmed through the measurement of correlation values. (2) Instead of using only one type of ant, an *inhomogeneous population* is used which moves at different speeds (uniformly distributed). Faster ants can skip several grid cells in one step. Each ant's perception is coupled to its speed, so that slower ants make "fussier" decisions about picking and dropping, i.e. the perception function becomes:

$$f(i) = max\left(0,\ \frac{1}{|S|} \sum_{\{c_{ij} \in S | d_{ij} \neq nil\}} \left(1 - \frac{d(k, d_{ij})}{\alpha\ \mu\ \frac{v-1}{V_{max}}}\right)\right)$$

When introducing inhomogeneous populations in isolation, our experiments do not reconfirm the performance gain reported in [LF94]. It appears that the improvements were due to introducing larger stepsizes simultaneously with inhomogeneous populations. We adopt the idea of short-term memory and use inhomogeneous populations with the minor but crucial modification of "jumps". We also introduce an adaptive scaling strategy and some further modifications to achieve reliable results and to improve the efficiency.

Adaptive scaling: The artificial test data used in the literature provide idealized conditions. Due to the choice of regularly distributed cluster centers and identical spread, inter-document distances are limited to a small range and smoothly distributed. When conducting experiments on more difficult test sets with irregular inter-cluster distances and standard deviations, more clusters and, in particular, of higher dimensionality, we found that an appropriate α value cannot be determined without a-priori knowledge of the data's structure. In

consequence, we introduce an adaptive strategy to determine the α value for a given data set. The sorting process starts with $\alpha = 0.1$ and increases α by 0.01 up to a maximum value of 1.0 after each sequence of 250 steps in which only few dropping or picking action occur.

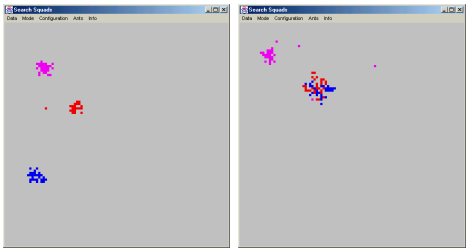
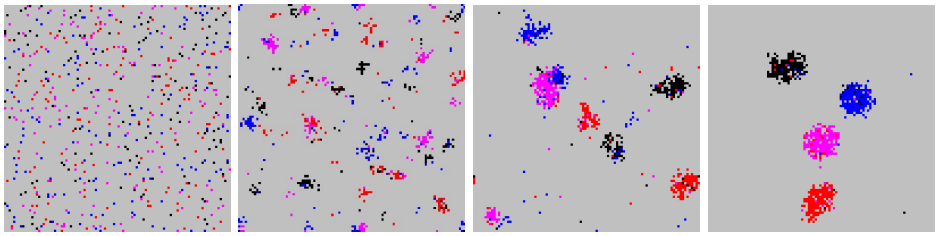


Fig. 4. Separation of Three Clusters (left: adaptive $\alpha \rightarrow 0.32$, right: non-adaptive $\alpha = 1.0$)

This means that our methods starts with a very fine distinction between data elements and reduces it only if necessary. In our experiments, this method reliably determined suitable alpha values. Also note that we scale similarity values in the interval $[1 - \frac{max_{i,j} d(i,j)}{\alpha\ \mu}, 1]$ instead of $[0, 1]$. Thus, *negative influence* of large dissimilarities is permitted which leads to improved cluster separation. Visual inspection of the sorting result for a three cluster data set with irregular inter-cluster distances in Fig. 4 shows that the adaptive strategy manages to deal well with it, whereas the non-adaptive strategy can only separate a single cluster.

Jumps: Experiments show that the use of inhomogeneous populations only leads to significant improvements in runtime and sorting quality if used with very large speed values (up to 50% of the grid size), as the ants' large steps favor the dissolution of preliminary small clusters. The interesting aspect here is that this introduces "jumping" ants and the smooth moving of an agent through a continuous space is transformed into a form of more global sampling with directional bias.

Stagnation control: With complex data, early stagnation of the whole clustering process can be a problem. This is caused by outliers in the data sets. Due to their high dissimilarity to all other data elements, agents do not manage to dispose off these items once they had been picked. This results in *blocked* ants performing random walks on the grid without contributing to the sorting process. Similar to [MSV99], we therefore use a *failure counter* for each ant. After 100 unsuccessful dropping attempts an ant drops its load regardless of the neighborhood's similarity.



(a) Initial scattering $t = 0$

(b) Original Alg. t=30, corr=0.01

(c) Original Alg. t=1000, corr=0.24

(d) Improved Alg. t=30, corr=0.6

Fig. 5. Results on four cluster test set (a,b: $\alpha = 1.0$, d: adaptive $\alpha \rightarrow 1.0$).

Eager ants: In the original algorithm, ants often spend large amounts of time searching for new documents to pick up. To prevent this time-consuming search we couple each ant with a new document immediately after it drops its load. As soon as this happens, the ant randomly chooses a document from the index of all non-carried documents, moves to its position and tries to pick it up. Failure results in the random choice of another document.

3.1 Evaluation

A comparison of the improved algorithm with the original version on the 4×200 test set introduced above shows a significant improvement of both sorting quality and time. As shown in Figure 5, both algorithms start at $t = 0$ with a random disorder of data elements on the grid. The slow clustering progress for the original algorithm can be observed in Figure 5b and Figure 5c. Only little clustering has been obtained after 30 iterations (one iteration consists of 10000 individual actions) and even for $t = 1000$ the four clusters within the data have not been correctly identified. The modified version, in contrast, reliably determines the four main clusters within the first 30 iterations. The corresponding spatial ordering of the test data can be seen in Figure 5d.

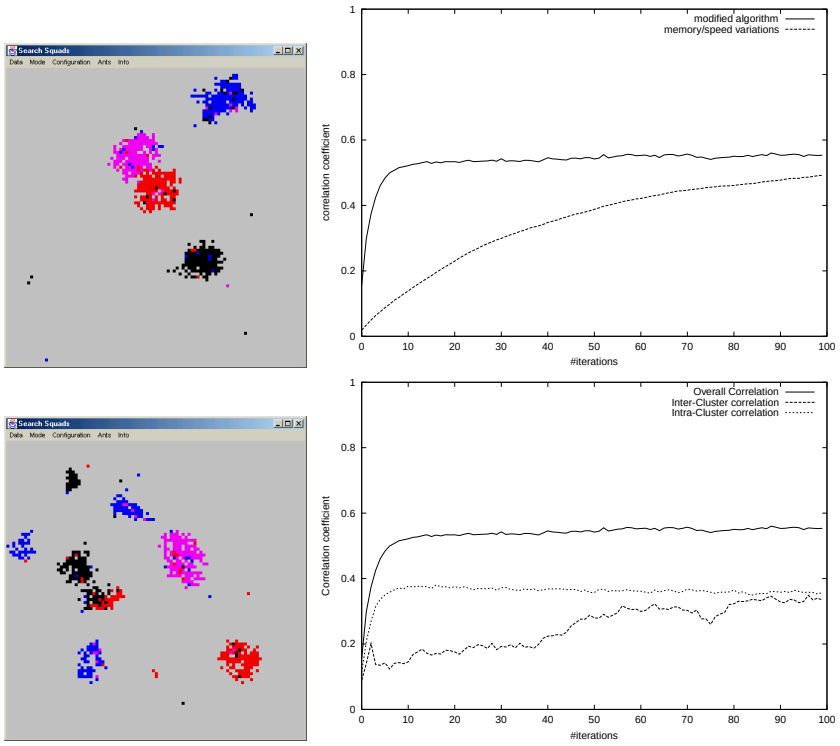


Fig. 6. Performance gain for all improvements (top left) and using only memory and inhomogeneous populations (bottom left). Overall correlation for both algorithms (top right) and individual correlations types for improved algorithm (bottom right).

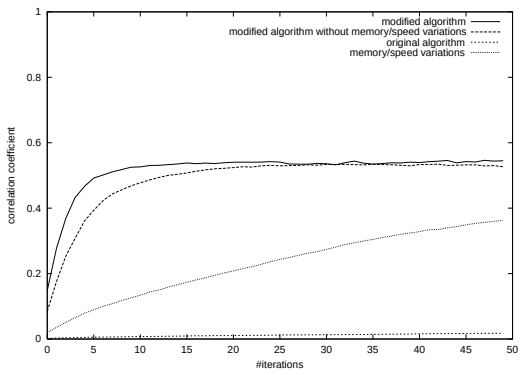


Fig. 7. Overall correlation for four versions of algorithms on 4×200 test data (mean values for 50 runs).

These differences in performance are reflected by the analytical performance measurements. While the modified ant-algorithm improves the overall correlation significantly and is soon close to convergence, the plots for the original algorithm show only little improvement in the same runtime (Fig. 7).

To further isolate the effect of our newly introduced modifications from that of memory and inhomogeneous populations alone, we compare them with the 800 element test set. Overall correlation as well as cluster identification are very clearly improved for the full version. Unfortunately, it also has to be noted that the achievable inter-cluster correlations are not superior (Fig. 6).

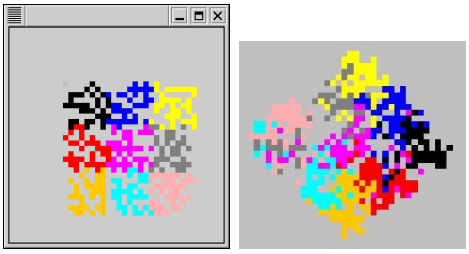


Fig. 8. Dense data: correlation=0.60

The quality of sorting if there are no distinct clusters in the data deserves individual assessment. For better visual observation we use a test set merely consisting of one cluster with uniformly distributed elements (Fig. 8). Nine different regions within the distribution are colored (left) so that the reconstruction of the order within the cluster can be observed (right). A visual inspection indicates that the algorithm is relatively successful in reproducing the approximate color ordering within the data. Exact measurement shows that the correlation for this test data rarely exceeds 0.7, but given that the goal of our algorithms is to generate embeddings for visual inspection only, it can be argued that this correlation is sufficient. This experiment suggests that it could be possible to achieve better intra-cluster correlations by adding a *second stage* of sorting in which ants with small step sizes are confined to move *within* each cluster.

3.2 Pre-processing

From above we see that the modified algorithm achieves reasonable overall correlations, but that the inter-cluster correlations are not sufficient for the purpose of topic-map generation. To obtain proper inter-cluster distances, we can exploit the fact that document positions are comparatively stable in regard to their initial positions. This allows us to initialize the clustering algorithm with an approximate (unclustered) embedding. While computing an optimal multi-dimensional scaling is a computationally hard problem, it is relatively easy to compute an approximate scaling that is sufficient for visual inspection. We do this with a straight-forward algorithm adopted from [NL01] which iteratively minimizes the sum-squared error between the distances d_{ij} in document space and the distances d'_{ij} in the two-dimensional map space updating the positions p_j via the

gradient descent rule: $p_{jk} \leftarrow p_{jk} - \frac{lr(d_{mj} - d'_{mj})}{d'_{mj}} abs(p_{jk} - p_{mk}) sign(p_{jk} - p_{mk})$ for each dimension ($k = 1, 2$) independently. lr is the learning rate (typically 0.05). The minimization stops when the increase of the Pearson correlation between d_{ij} and d'_{ij} falls under a pre-defined threshold. Clustering starts from the approximate scaling, which results in significantly improved correlation values of the final embedding. For the four-cluster test set an average inter-cluster correlation of almost 0.9 is obtained, improving the overall correlation to about 0.7.

4 Related Work

The first ant-based sorting algorithm was presented by Deneuborg [Den90] and was designed for collaborative robotics. In this work, only binary classification was addressed. Lumer and Faeita [LF94] later extended these ideas to clustering more complex data sets based on continuous similarity functions. Modified versions of this algorithm have later been applied to graph clustering, focusing in particular on applications in VLSI [KS94,KLS97,KS99,KSL98]. For an extensive survey of different ant-based meta-heuristics which are based on pheromone-trail communication, the interested reader is invited to refer to [CDG99].

The idea of using a map metaphor for visualizing contents similarity has some tradition and a full survey is beyond the scope of this paper. The idea was originally introduced by Chalmers [Cha93] and later picked up in other projects, among them the commercial systems Cartia (`www.aurigin.com`), Spire (`showcase.pnl.gov`) and Kartoo (`www.kartoo.com`), Lighthouse [LA00] and others. A recent HCI study clearly shows that a landscape metaphor can help to significantly enhance user performance in search tasks [Fab00]. A significant proportion of systems that generate such visualization has relied on self-organizing maps. Since generating topic maps with self-organizing maps requires computation times that are prohibitive in an interactive setting [Lag00], we decided to investigate ant-based sorting/clustering as an alternative. To the best of our knowledge, this is the first system to generate large topic maps for several thousand documents in an interactive setting. Ant-based clustering for a map of 4200 documents, for example, takes 29 seconds on a 990MHz Pentium-III running JDK 1.4.1 and would be considerably faster in an optimized C implementation.

5 Conclusions

We have re-examined ant-based sorting and clustering methods in the context of a real-life application and demonstrated modifications of the algorithms that yield significant improvements in terms of quality and speed.

It is obvious that we have sacrificed most of the original methods' biological plausibility for the performance improvements. However, this should not concern us, as we are not trying to analyze the behavior of real insects, rather we use their behavior as inspiration for the design of an efficient meta-heuristics. In fact, lifting the restrictions imposed by real physical systems (and therefore sacrificing the biological plausibility) leads to some interesting insights: for example, an intuitive explanation of why ant-based sorting works well may assume that the coherence of the space in which the ants operate is crucial to the function of the algorithms. As it turns out with the introduction of jumps, this is not the case.

The use of a stochastic algorithm in a query system poses challenging questions concerning query refinement (incrementallity) and query repetition (stability) which we are planning to investigate. Our evaluation of the algorithm was mainly based on measurements for artificial test data and on visual observation for real query data. Clearly a thorough evaluation for real test data and user studies are important question to be addressed in the future.

References

[CC94] T.F. Cox and M.A.A. Cox. *Multidimensional Scaling.* Chapman & Hall, 1994.

[CD00] Hao Chen and Susan Dumais. Bringing order to the web. In *ACM CHI*, The Hague, April 2000.

[CDG99] D. Corne, M. Dorigo, and F. Glover, editors. *New Ideas in Optimization*, chapter 2: The Ant Colony Optimization Meta-Heuristic, pages 379–387. McGraw-Hill International (UK) Limited, 1999.

[Cha93] M. Chalmers. Using a landscape metaphor to represent a corpus of documents. In A. Frank and I. Campari, editors, *Spatial Information Theory: A Theoretical Basis for GIS*, pages 377–390. Springer-Verlag, September 1993.

[DDL+90] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[Den90] J. L. Deneuborg. The dynamics of collective sorting. robot-like ants and ant-like robots. In *1st International Conference on Simulation of Adaptive Behaviour: From animals to animats 1*, pages 356–363. MIT Press, Mai 1990.

[Fab00] S. I. Fabrikant. *Spatial Metaphors for Browsing Large Data Archives.* PhD thesis, Department of Geography, University of Colorado, 2000.

[Han] J. Handl. Visualising internet-queries using ant-based heuristics. Honours Thesis. Dept. of Computer Science, Monash University, Australia. 2001.

[KLS97] P. Kuntz, P. Layzell, and D. Snyers. A colony of ant-like agents for partitioning in VLSI technology. In *4th European Conference on Artificial Life.* MIT Press, July 1997.

[KS94] P Kuntz and D. Snyers. Emergent colonization and graph partitioning. In *3rd International Conference on Simulation of Adaptive Behaviour: From Animals to Animats 3.* MIT Press, April 1994.

[KS99] P. Kuntz and D. Snyers. New results on an ant-based heuristic for highlighting the organization of large graphs. In *99 Congress on Evolutionary Computation*, pages 1451–1458. IEEE Press, July 1999.

[KSL98] P. Kuntz, D. Snyers, and P. Layzell. A stochastic heuristic for visualising graph clusters in a bi-dimensional space prior to partitioning. *Journal of Heuristics*, 1998.

[LA00] A. Leuski and J. Allan. Lighthouse: Showing the way to relevant information. In *IEEE Information Vizualization*, Salt Lake City, October 2000.

[Lag00] K. Lagus. *Text Mining with the WEBSOM.* PhD thesis, Department of Computer Science and Engineering, Helsinki University of Technology, 2000.

[LF94] E. Lumer and B. Faieta. Diversity and adaption in populations of clustering ants. In *3rd International Conference on Simulation of Adaptive Behaviour: From Animals to Animats 3.* MIT Press, July 1994.

[MSV99] N. Monmarche, M. Slimane, and G. Venturini. On improving clustering in numerical databases with artificial ants. In *Advances in Artificial Life (ECAL'99), LNAI 1674.* Springer-Verlag, 1999.

[NL01] D. J. Navarro and M. D. Lee. Spatial visualisation of document similarity. In *Defence Human Factors Special Interest Group Meeting*, August 2001.

[Sal88] G. Salton. *Automatic Text Processing.* Addison-Wesley, New York, 1988.

[ZE99] O. Zamir and O. Etzioni. Grouper: A dynamic clustering interface to web search results. In *8th World Wide Web Conference*, Toronto, May 1999.