An Improved Probabilistic Ant based Clustering for Distributed Databases

R.Chandrasekar[1] **T.Srinivasan**[2]
[1]Department of Information Technology
[2]Assistant Professor, Department of Computer Science and Engineering
Sri Venkateswara College of Engineering, Sriperumbudur, India
[1]chandra85@gmail.com, [2]tsrini1969@gmail.com

Abstract

In this paper we present an improved version of the Probabilistic Ant based Clustering Algorithm for Distributed Databases (PACE). The most important feature of this algorithm is the formation of numerous zones in different sites based on corresponding user queries to the distributed database. Keywords, extracted out of the queries, are used to assign a range of values according to their corresponding probability of occurrence or hit ratio at each site. We propose the introduction of weights for individual or groups of data items in each zone according to their relevance to the queries along with the concept of familial pheromone trails as part of an Ant Odor Identification Model to bias the movements of different types of ants towards the members of their own family. Its performance is compared against PACE and other known clustering algorithms for different evaluation measures and an improvement is shown in terms of convergence speed and quality of solution obtained.

1 Introduction

Swarm systems have recently become a source of inspiration for the design of various clustering algorithms in [Lumer and Faieta, 1994; Handl *et al.*, 2003a; Handl *et al.*, 2003b; Handl *et al.*, 2005]. Ant based clustering and sorting was first introduced by [Bonabeau *et al*, 1999] to explain different types of naturally occurring emergent phenomena. It is an instance of the broad category of ant algorithms [Dorigo *et al*, 1999; Dorigo *et al.*, 2000]; that is, algorithms that model 'some behavior' observed in real ants. In the case of ant-based clustering and sorting, two related types of natural ant behavior are modeled. While the traditional ant based algorithms have described clustering data in a single site, we focus on the clustering mechanisms for data in distributed sites [Johnson and Kargupta, 1999]. For example, we may have a number of banks belonging to a multinational banking chain, and each bank maintains a database describing its members. Then we may cluster the databases to learn new high-level concepts that characterize groups of banks. Rather than restricting learning to specific databases, with increasingly more databases becoming available on the Internet, we can globalize knowledge discovery and learn general patterns by this approach [Forman and Zhang, 2000].

The use of ant based clustering for distributed databases was explored in [Chandrasekar *et al*, 2006]. An algorithm called probabilistic ant based clustering for distributed databases (PACE) was proposed based on user-interaction or queries from the distributed database. The main advantage is that highly probable or most likely keywords from the query can be further analyzed instead of concentrating on the entire set of data available. It utilizes a commonly heard concept of hit ratio, which it calculates for the user query. Depending on this, a number of zones are formed throughout the database with priorities assigned to them. The sizes of these zones and their logical placement in the database are discussed in that paper. A colony building algorithm of ants is utilized for formation of the clusters with an extensive odor analysis model which determines the number and type of agents or ants surrounding a data item. The results obtained from PACE showed highly efficient retrieval from the final clusters formed. In spite of this there are many shortcomings with PACE. The convergence time for the algorithm is not shown to be quick. This is necessary because with increasing sizes of the database, though not shown, the entire process may take a correspondingly large time. Many parameters were not functions of the sizes of the data set to improve the convergence time. Also no further information could be gathered from the final solution obtained other than which points lay in what clusters according to the user queries. Rest of the paper is organized as follows; Section 2 describes the main motivations for our work, Section 3 gives an overview of PACE, Section 4 gives the proposed improvements, Section 5 some Experimental Results while Section 6 concludes our work.

2 Main Motivations for Our Work

Our work has been inspired by the rules specified in [Johnson and Kargupta, 1999]. The time duration of the algorithm should be less than or equal to that of clustering from a sin-

gle site considering all the data to be present in a single site . To wait for each and every zone to complete its intra-zone clustering and then to perform inter-zone clustering would entail a substantial increase in time. To reduce this considerably, we propose the parallelization of the clustering algorithm so that inter-zone clustering can take place as and when zones become free after intra-zone clustering. There should also be a metric of dissimilarity between points and clusters obtained in the final solution. By the introduction of weights in our algorithm, the final heaps formed by the ants could be in a weighted order. A reasonably informative metric of dissimilarity could be a range of weights w_{min} and w_{max} between any two points which is more suitable for clusters formed depending on the user queries to the distributed database rather than calculating the minimum and maximum possible distances between any two points in the final tree. Data gathered from different sites should also not be transmitted to a single site for processing. To circumvent this, building of a global model from a number of local models was suggested in [Johnson and Kargupta, 1999]. Instead of this, the presence of well-defined zones and with parallelization of clustering reduces the transmission costs involved rather than transmitting data to a single site.

For our proposed improvements to PACE, data items in a zone-even after inter-zone clustering has taken place-are marked in a scale either individually or as a group with a weight, in ACO terms-a distinct odor, to denote their relevance to the user queries to the distributed database. If the odor associated with that group is strong then their associated weights are also correspondingly large and hence it makes sense to perform clustering in that region of data items first. This partially satisfies the criterion of a fast convergence time and this along with the idea the notion of parallelization of clustering between the zones provides a better performance than PACE as we show in our results. Another important improvement is the introduction of familial pheromone trails. A family of ants follows its own distinctive trails unless in some species, for example, a trait of trying to impersonate other species exists. This holds true even for a distinctive odor possessed by each and every species [Bonabeau *et al*, 1999] which formed the basis of identifying the number and type of agents surrounding a data item in PACE. We utilize this concept to enable the formation of clear and defined clusters by allowing each family or species of ants to follow their own familial trails. As we show later, this proves to be effective for different densities of data items. For the re-ordering and agglomeration each time after intra-zone and inter-zone clustering has been performed, it makes sense to re-order them based on those weights so that the quantities w_{min} and w_{max} can be calculated easily for effective representation to the user.

3 Overview of PACE Algorithm

From Chandrasekar [2006] it can be seen that the query from the user is disintegrated into atomic keywords each of which can be identified uniquely from the databases at various sites. The number of occurrences of each keyword in the sites is computed using a *hit ratio*.

$$H(r_m) = \sum_{d_m=1}^{d_m=m} \sum_{i=1}^{i=n} \frac{1}{k_i} \tag{1}$$

The variable K represents the key words and the limit of the summation is '*n*' which is equal to the number of keywords. Using this hit ratio each of the distributed sites is assigned a probability value,

$$P(d_m) = 1 - H(r_m) \tag{2}$$

The data sites are then ordered in descending order of their probabilities. Each data site is then divided into numerous bounded regions called zones. The agents (ants) are distributed across various zones. Clustering at each stage is achieved by agglomeration and sorting, which is performed in an iterative manner.

An ant in real situation identifies other group of ants of their same colony using a distinctive odor possessed by them that is unique to each colony [Bonabeau *et al*, 1999; Dorigo *et al*, 1999; Dorigo *et al.*, 2000].This behavior of ants is exploited in PACE to identify and form a group of ants that carry related data. Again the ants in a real situation move freely in search of food sources. Whenever they find a food source, they cluster around it from Chandrasekar [2006]. Similar behavior is incorporated in PACE. Since the ants are free to move within their zones, they tend to move in search of data objects (Food). In doing so, ants tend to cluster around various data objects. The cluster of ants around groups of data objects hence forms a Family of Agents (FoA). In this way group of ants clustering around various data objects within different parts of a zone form numerous FoA. Thus the ants within a FoA begin to build their colony using the data that they have collected. This is called as an Ant Odor Identification Model (AOIM) which is extended in our work.

Once the FoA is formed, the ants begin to carry data items specific to their family (colony) according to the Picking (P_p) and Dropping (P_d) probabilities [Lumer and Faieta, 1994; Handl *et al.*, 2003b] which depend on the environment surrounding the ants. In PACE, heaps formed are sorted so that the data items closer to the keywords in terms of relevance are placed so that they can be easily retrieved. Every FoA within the zones form their own colony and repeat the procedure. This is called as an *Ant Colony Building Algorithm* (*ACB*). The colonies that are built form the primary clusters. The final step would be to perform agglomeration. PACE uses complete-link type of agglomeration. This method assumes that all data items in the cluster are very similar to each other. This is possible because of the sequence of execution from intra-zone to inter-zone agglomeration.

These clusters are agglomerated through three different stages before the final cluster is formed. First is the *Intra-Zone Agglomeration*, where the primary clusters within the zone are agglomerated and then reordered by sorting. At the end of this stage a single cluster is obtained. Next is the *Inter-Zone Agglomeration*. In this stage the clusters from each zone are agglomerated to form a single cluster for the entire data site. In this way many clusters are obtained each data site. Again sorting of the cluster is performed to retain the data items with maximum relevance to the keywords in a way which can be easily retrieved. The final stage involves agglomerating the clusters from the distributed data sites to form a single final cluster. This cluster again is sorted and contains the best results corresponding to the user query.

4 Proposed Improvements

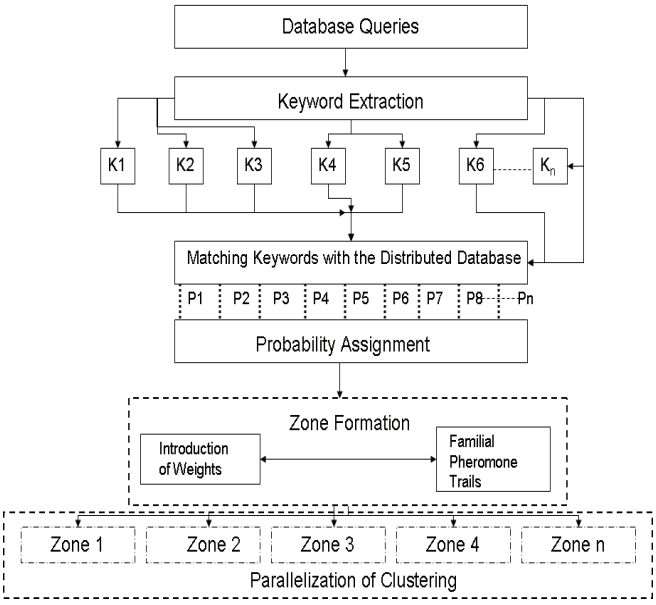

Figure 1: Architecture of the Improved Probabilistic Ant based Clustering for Distributed Databases

4.1 Introduction of weights

Depending on the hit ratio, individual or groups of data items are marked on a scale denoting their relevance to the query, called as d_w or weights. d_w denotes the odor associated with an item or a group of items surrounding it. If the odor is strong, then the associated weights are strong and hence it makes sense to perform clustering there first. There is an inherent relationship between the hit ratio and d_w denoted by the following equation which holds true for both individual or groups of data items,

$$d_{wi} = H(r_m)\sum_{j=1}^{m} R_j \qquad (3)$$

Where R is the radius of a zone and $H(r_m)$ is the hit ratio for the overall database. We further analyze the memory costs of allocating weights for individual or groups of data items. For individual data items, the memory costs may grow linearly with corresponding increase in data items. As an alternative, groups of data items which either partially or completely satisfies the queries can be marked with an odor- in this case a collective weight or a *collective odor*-to eliminate the need for maintaining separate weights for the data items. This could be the case when the density of data items in a region on the toroidal grid is not too high. Traditionally ant based clustering algorithms have employed some kind of a 'look-ahead' memory [Handl *et al.*, 2003b; Handl *et al.*, 2003a] which is used to bias the search of the ants towards recently visited regions with a strong neighborhood function. From PACE, the neighborhood functions[Lumer and Faieta, 1994] can be obtained as:

Case 1:

$$1-\frac{\Delta(i,j)}{\alpha''}, \text{ Where } \alpha'' = \alpha^3, 0.7 < H(r_m) < 1$$

Case 2:

$$1-\frac{\Delta(i,j)}{\alpha'}, \text{ Where } \alpha' = \alpha^2, 0.4 < H(r_m) < 0.7$$

Case3:

$$1-\frac{\Delta(i,j)}{\alpha}, \quad 0.1 < H(r_m) < 0.4$$

Case 4:

0, Otherwise

The power of α is derived experimentally. We modify the neighborhood function of PACE and other ant based clustering algorithms as,

$$f(i) = \max(0.0, \frac{1}{\sigma^2} \cdot d_w \sum_{jEL} 1-\frac{\Delta(i,j)}{\alpha}) \qquad (4)$$

We include d_w in the neighborhood function as a measure of the weight associated with a particular region. This can be modified to include d_w for either individual or groups of data-items depending on the density of the region as,

$$f(i) = \max(0.0, \frac{1}{\sigma^2} \cdot \sum_{i=1}^{n} d_{wi} \sum_{jEL} 1-\frac{\Delta(i,j)}{\alpha}) \qquad (5)$$

Whenever *n* is the number of data items. By doing either or all of the above, the requirements of fast convergence are satisfied as the clusters are formed in the decreasing order of the regions with weights. This being so because ants get

attracted to a stronger odor more easily than a weak one. So clusters tend to be formed first in those regions with a strong odor and then the remaining thus forming heaps which are more relevant to the user queries; the latter themselves being a standard set of queries. For Inter-Zone clustering, an average weight is associated with the clusters and the same procedure is repeated as above.

4.2 Familial Pheromone Trails

A Family of Ants which follows their own pheromone trails form clearly defined clusters. Instead of modifying the P_{pick} and P_{drop} functions, data items picked according to d_w are dropped around regions satisfying both FoA and d_w in some measure. We build upon the concepts of PACE and in addition to defining a FoA as a family which is formed around groups of items, we define each FoA as to having its unique pheromone trails. As we shall show later, the effect of this is control over the definitions of-or demarcations between-the clusters formed. If more priority is given to d_w then ants may pick up and drop items near highly odoriferous regions leading to overlapping or fused clusters. Alternatively if more priority is given to an Ant's FoA then data items may be grouped together which are totally unrelated to each other in certain cases or to the user queries in general. The relationship between FoA and d_w is as follows,

$$FoA = \left(\frac{AOIM_i}{AOIM_j} \right) \cdot d_w \tag{6}$$

The main advantage of choosing a proper balance between FoA and d_w is for choosing clearly defined and compact clusters. By allowing for different groups of ants each part of their own FoA, considerable spreading of data in the grid takes place. Each group of ants lies separately in the grid without any overlapping of groups. To dampen the effects of outliers, which are points of non-agglomerative behavior, we follow a two- stage strategy.

In the first stage, during the ant colony building algorithm, ants tend to cluster around their own families-surrounding groups of data items-hence isolated data items are those not having a strong odor and tend to appear at the bottom of the heaps formed by the ants. By this, an iterative agglomeration and re-ordering renders these data items in a way that they do not appear earlier to the users than more relevant data items. This re-ordering takes place more frequently after a number of clusters which have been merged together reduces the clusters to below a certain threshold, typically 1/3 of the original number. In the second stage during the final clustering steps when the granularity [Jain and Dubes, 1988;Jain *et al*, 1999] is very high, again the re-ordering is performed so that the irrelevant data items and their clusters, generally very small, are below the other clusters.

4.3 Relationship of Radius of Perception with Zone Radius and d_w

The radius of perception is given by (σ-1)/2 [Lumer and Faieta, 1994] for the general neighborhood function. For our algorithm it depends on the Zone Radius and d_w as follows,

$$Rp = R \cdot \sum_{m=1}^{R} d_{wm} \tag{7}$$

Where Rp is the radius of perception and R the zone radius of that zone where the agent is currently placed. Increasing the radius of perception provides a better range for the agents to choose from but increases costs. To prevent this, we propose to initially allocate the Radius of Perception depending on the Zone Radius, increase it 'locally' for each ant depending on its environment and the values of d_w. Thus it is kept as a parameter of the dataset as required by [Johnson and Kargupta, 1999]. There is no real need to modify the neighborhood and threshold functions, P_{pick} and P_{drop} . Instead of the former, spatial separation between the clusters is maintained throughout by the introduction of weights and the familial pheromone trails. Instead of the latter, done to speed up the clustering process, biasing the movements of ants towards their own families and according to the distribution of weights provides a faster convergence time (See Section 4.1).

4.4 Weighted Agglomeration and Parallel Clustering

As mentioned earlier, the final heaps formed are arranged in a weighted order. For deriving information other than which points lie in what clusters, we use two values w_{min} and w_{max} as measures of dissimilarity between any two points in the heap. They are as follows,

$$w_{\min} = Min(w1, w2) \tag{8}$$

$$w_{\max} = Max(w1, w2) \tag{9}$$

w_{min} is a measure of the minimum difference in weights between any two points. Since the values are absolute and defined for every leaf node of the final heap, there is no need of explicitly calculating it every time like in [Johnson and Kargupta, 1999] where the minimum and maximum distances between any two points in a dendogram were calculated frequently from the final solution. Similarly w_{max} is a measure of the maximum difference in weights between any points and the same principles as w_{min} apply to it also. To save on processing time and to speed up the clustering, we have proposed the parallelization of the agglomeration process. Intra-zone agglomeration takes place simultaneously in the various sites starting from zones of higher priority as in PACE. Inter-zone agglomeration takes place as and when any zone finishes its own intra-zone process. Thus zones in these sites need not wait until all other zones in the

distributed database have finished-thus speeding up the agglomeration process.

5 Experimental Results

In this section comparative results of the performance of our improved PACE algorithm (denoted as *I-PACE*) are reported against established clustering algorithms like the k-means algorithm, the hierarchical agglomerative algorithm and PACE.

5.1 Experimental Setup

The evaluation methodology given here was inspired by [Handl *et al.*, 2003a; Handl *et al.*, 2003b].The first algorithm we compare against is the well-known k-means algorithm. Random initialization is used is used in our experiments, and the best result out of a few runs is selected. If empty clusters arise during the clustering process using the k-means algorithm [Bishop, 1995], these are reinitialized using a randomly selected data item. As a second method, an agglomerative hierarchical clustering algorithm [Day and Edelsbrunner, 1984] based on the linkage metric *average link* [Jain *et al*, 1999] is used. The data items are distributed in a number of toroidal grids with the assumption that the ants are able to move from one grid to another and that the edges are spherical to enable movements from all sides.

Synthetic Data

The *Square1* dataset is the most popularly used type of data set used for evaluation purposes in ant based clustering. It is two-dimensional and consists of four clusters arranged as a square. To conform to distributed datasets, the *Square1* dataset is spread uniformly among the various sites. They are generated by the Normal Distributions with corresponding points as (N(-5,2),N(-5,2)),(N(5,2),N(5,2)), (N(-5,2),N(5,2)) and (N(5,2),N(-5,2)) and are each of size 200.

Real Data

The real data collections used were the *Iris* data, the *Wine Recognition* data and the *Yeast* data with the description of each collection in [Blake and Merz, 1998]. Each dataset is permuted and randomly distributed in the sites. Different evaluation functions proposed in [Handl *et al.*, 2003a] are adapted for comparing the clustering results obtained from applying the three clustering algorithms on the test sets. The *F-Measure* [Rijsbergen, 1979]*, Dunn Index* [Halkidi *et al.*, 2000] *and Rand Index* [Rijsbergen, 1979] are the three measures and their respective definitions also mentioned in [Handl *et al.*, 2003b] and each should be maximized.

5.2 Experimental Results

The results of the evaluation functions on both synthetic as well as the real data are shown in Table 1. It shows the mean and standard deviation (obtained over 60 runs) for each of the three measures. The above results demonstrate that in spite of clear cluster structures not being present in certain data, I-PACE is quite easily able to identify the correct number of clusters. With certain exceptions I-PACE performs well for all the three evaluation functions. Though k-means shows better performance for the synthetic data, it is clearly outmatched with respect to the real data by I-PACE. Errors in each case for the I-PACE algorithm are also considerably smaller than the remaining three as shown here in Tables 2, 3 and 4 averaged over 80 runs.

TABLE I
RESULTS OF EVALUATION FUNCTIONS ON K-MEANS, HIERARCHICAL AGGLOMERATIVE AVERAGE LINK CLUSTERING, PACE AND I-PACE.THE TABLE SHOWS MEAN AND STANDARD DEVIATIONS (IN BRACKETS) FOR 60 RUNS

Square1	k-means	Average Link	PACE	I-PACE
#Clusters	4(0)	4(0)	4(0)	4(0)
F-Measure	**0.9871**(0.004)	0.9806(0.0076)	0.9812(0.00452)	0.9792(0.00552)
Dunn Index	3.678(0.0998)	3.21113(0.287)	3.68765(0.234)	**3.88765**(0.00021)
Rand Index	0.987(0.0035)	0.9799(0.0069)	0.98234(0.0045)	**0.98934**(0.00035)
IRIS	k-means	Average Link	PACE	I-PACE
#Clusters	3(0)	3(0)	3(0)	3(0)
F-Measure	0.811(0.0762)	0.811887(0.0)	0.82245(0.0135)	**0.83245**(0.00012)
Dunn Index	2.64(0.41009)	2.4588(0.0)	2.98566(0.2337)	**3.18566**(0.0024)
Rand Index	0.810(0.1004)	**0.8314**(0.0)	0.8237(0.00819)	0.83(0.00082)
WINE	k-means	Average Link	PACE	I-PACE
#Clusters	3(0)	3(0)	3(0)	3(0)
F-Measure	0.8217(0.034)	0.84232(0.0)	0.87711(0.0034)	**0.89915**(0.00021)
Dunn Index	2.789(0.3458)	2.655478(0.0)	**2.998**(0.008766)	2.898(0.00866)
Rand Index	0.819(0.0901)	0.81244(0.0)	0.822(0.0009911)	**0.829**(0.00089911
YEAST	k-means	Average Link	PACE	I-PACE
#Clusters	10(0)	10(0)	10(0)	10(0)
F-Measure	0.4221(0.003)	**0.452234**(0.0)	0.44489(0.0899)	0.43489(0.0899)
Dunn Index	1.7234(0.102)	1.60023(0.0)	1.7876(0.20034)	**1.8876**(0.00004)
Rand Index	0.7432(0.001)	0.742234(0.0)	**0.7892**(0.08997)	**0.7992**(0.00095)

TABLE II
PARAMETERS AND TEST RESULTS SHOWING THE CLUSTERING QUALITY WITH RESPECT TO ERRORS ON THE IRIS DATA

Parameter	K-Means	Average Link	PACE	I-PACE
Minimum Errors	3	4	2	1
Maximum Errors	8	7	4	3
Average Errors	3.33	2.57	1.46	1.1
Percentage of Errors	4.67%	3.89%	1.78%	1%

TABLE III
PARAMETERS AND TEST RESULTS SHOWING THE CLUSTERING QUALITY WITH RESPECT TO ERRORS ON THE WINE RECOGNITION DATA

Parameter	K-Means	Average Link	PACE	I-PACE
Minimum Errors	4	2	1	1
Maximum Errors	7	9	4	3
Average Errors	4.33	2.37	1.46	1.2
Percentage of Errors	4.97%	3.19%	1.98%	1.1%

TABLE IV
PARAMETERS AND TEST RESULTS SHOWING THE CLUSTERING QUALITY WITH RESPECT TO ERRORS ON THE YEAST DATA

Parameter	K-Means	Average Link	PACE	I-PACE
Minimum Errors	6	4	2	2
Maximum Errors	9	6	5	3
Average Errors	6.33	2.37	2.46	1.5
Percentage of Errors	5.97%	4.19%	2.18%	1%

6 Conclusion

In this paper we have proposed an improved version of the PACE algorithm for ant-based clustering in distributed databases. The main features of this algorithm are the introduction of weights and familial pheromone trails as part of an Ant Odor Identification Model. The aims are manifold-to reduce the convergence time and thereby improve the runtime of the solution, to improve the quality of clustering by forming compact and clearly defined clusters, to separate out outliers from the final solution, to provide a metric which easily determines the dissimilarity between any two points in the final cluster, to parallelize the algorithm for again reducing the processing time. These aims are reasonably achieved by our algorithm as ably proven by our experimental results. Further investigation could be carried out with a large number of data-sets as an extended version of our algorithm.

References

[Bonabeau *et al*, 1999] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence – From Natural to Artificial Systems*. Oxford University Press, New York, NY, 1999.

[Blake and Merz, 1998] Blake, C. L. and Merz, C. J.UCI Repository of machine learning databases [http://www.ics.uci.edu/mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science, 1998

[Lumer and Faieta, 1994] Lumer E D and Faieta B. Diversity and Adaptation in Populations of Clustering Ants. Cli D, Husbands P, Meyer J and Wilson S (Eds.), *Proceedings of the Third International Conference on Simulation of Adaptive Behaviour: From Animals to Animats 3*, Cambridge, MA: MIT Press, 501-508.

[Dorigo *et al*, 1999] M. Dorigo, G. Di Caro, and L. Gambardella. Ant Colony Optimization: A New Metaheuristic. In Peter J. Angeline, Zbyszek Michalewicz, Marc Schoenauer, Xin Yao, and Ali Zalzala, editors, *Proceedings of the Congress on Evolutionary Computation,* volume 2, pages 1470--1477. IEEE Press, 1999.

[Chandrasekar *et al*, 2006] R.Chandrasekar,Vivek Vijaykumar and T.Srinivasan. Probabilistic Ant based Clustering for Distributed Databases. *In Proc. IEEE International Conference on Intelligent Systems 2006*, London,UK,September 2006.

[Johnson and Kargupta, 1999] Erik L. Johnson, Hillol Kargupta, Collective, Hierarchical Clustering from Distributed, Heterogeneous Data. *Large-Scale Parallel KDD Systems*, Eds. Zaki M. and Ho C., LNCS 1759, Springer-Verlag, Pages 221-244, 1999.

[Dhillon and Modha, 1999] Dhillon,I.and Modha, D. A Data-Clustering Algorithm on Distributed Memory Multiprocessors. *In Proceedings of the Workshop on Large-Scale Parallel KDD Systems,* ACM Press, New York, NY, 1999.

[Forman and Zhang, 2000] G. Forman and B. Zhang, Distributed Data Clustering Can be Efficient and Exact. *Proc. SIGKDD Explorations*, vol. 2, no. 2, Dec. 2000

[Deneubourg *et al*, 1990] J.-L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, and L. Chretien. The Dynamics of Collective Sorting: Robot-like Ant and Ant-like Robot. In J.-A. Meyer and S. W. Wilson, editors, *From Animals to Animats. Proceedings of the First International Conference on Simulation of Adaptive Behavior (SAB90)*, pages 356--365. MIT Press, Cambridge, MA, 1990

[Dorigo *et al.*, 2000] M. Dorigo, E. Bonabeau, and G. Theraulaz. Ant Algorithms and Stigmery. *Future Generation Computer Systems*, 16(8):851–871, 2000.

[Jain and Dubes, 1988] A.K. Jain, R. Dubes. *Algorithms for Clustering Data*, Prentice Hall, Englewood Cliffs, NJ, 1988.

[Jain *et al*, 1999] A. K. Jain, M. N. Murty, and P. J. Flynn. Data Clustering: A Review. *ACM Computing Surveys*, vol.31, no. 3, pp. 264-323, 1999.

[Handl *et al.*, 2005] Handl, J., Knowles, J., and Dorigo, M. Ant-based Clustering and Topographic Mapping. *Artificial Life* 12(1), 2005.

[Handl *et al.*, 2003a] Handl, J., Knowles, J., and Dorigo, M. On the Performance of Ant-based Clustering. *In Design and Application of Hybrid Intelligent Systems*, Vol. 104 of Frontiers in Artificial Intelligence and Applications (pp. 204-213). Amsterdam, The Netherlands: IOS Press.

[Handl *et al.*, 2003b] Handl, J., Knowles, J. and Dorigo, M. Ant-based clustering: a comparative study of its relative performance with respect to k-means, average link and 1D-som. Technical Report TR/IRIDIA/2003-24. IRIDIA, Universite Libre de Bruxelles, Belgium, 2003.

[Rijsbergen, 1979] Rijsbergen, C. V. *Information Retrieval*, 2nd edition. London, UK: Butterworths,1979.

[Halkidi *et al.*, 2000] Halkidi, M., Vazirgiannis, M., & Batistakis, I. (2000). Quality scheme assessment in the clustering process. *In Proceedings of the Fourth European Conference on Principles of Data Mining and Knowledge Discovery*, Vol. 1910 of Lecture Notes in Computer Science (pp. 265–267). Heidelberg, Germany: Springer-Verlag.

[Day and Edelsbrunner, 1984] William H. Day and Herbert Edelsbrunner. 1984. Efficient Algorithms for Agglomerative Hierarchical Clustering Methods. *Journal of Classification*. Volume 1, pp. 1-24.

[Bishop, 1995] Bishop, C. M. *Neural Networks for Pattern Recognition.* Oxford, England: Oxford University Press, 1995.