Effects of Inter-agent Communication in Ant-Based Clustering Algorithms: A case Study on Communication Policies in Swarm Systems

Marco A. Montes de Oca, Leonardo Garrido, and José L. Aguirre

Centro de Sistemas Inteligentes Tecnológico de Monterrey,
Campus Monterrey, Eugenio Garza Sada 2501,
C.P. 64849. Monterrey, México
m.montes@exatec.itesm.mx,
{leonardo.garrido, jlaguirre}@itesm.mx

Abstract. Communication among agents in swarm intelligent systems and more generally in multiagent systems, is crucial in order to coordinate agents' activities so that a particular goal at the collective level is achieved. From an agent's perspective, the problem consists in establishing communication policies that determine *what*, *when*, and *how* to communicate with others. In general, communication policies will depend on the nature of the problem being solved. This means that the solvability of problems by swarm intelligent systems depends, among other things, on the agents' communication policies, and setting an incorrect set of policies into the agents may result in finding poor solutions or even in the unsolvability of problems. As a case study, this paper focus on the effects of letting agents use different communication policies in ant-based clustering algorithms. Our results show the effects of using different communication policies on the final outcome of these algorithms.

1 Introduction

The term *Swarm Intelligence* is used to denote the relatively new discipline that studies systems that exhibit self-organizing properties at the global level from interactions of their lower level components. These studies are often inspired by the observation of social insects and other animal societies [1]. We will refer to systems with these features as *swarm intelligent systems*.

For them to work, swarm intelligent systems need the interaction of their constituent entities. In natural settings, *stigmergy* [6] plays a key role as it provides the means for indirect communication among insects through the environment. This same phenomenon has been successfully exploited in many systems used to solve combinatorial optimization problems [4], in clustering algorithms [7], and in robotic systems [10]. In spite of these successful experiences, we need to consider the question of whether agents should/could communicate in other ways to achieve organization or better solutions to problems. There is no general answer nor general communication policy that will apply equally well to all problems. This is why we think we need to study the effects of letting agents use different

A. Gelbukh, A. de Albornoz, and H. Terashima (Eds.): MICAI 2005, LNAI 3789, pp. 254–263, 2005.

communication policies. With this knowledge and considering the characteristics of a particular problem, we could either improve the performance of a swarm intelligent system or permit its application to a problem that was not possible before. This is the main motivation of our work.

2 Communication Policies Among Agents in Swarm Intelligent Systems

The collective behavior of social insects and other animal societies has inspired the design of metaheuristics that have found their first applications in the field of optimization [4, 11]. One of the first of such metaheuristics is Ant Colony Optimization, or ACO for short [3]. In ACO, a colony of artificial ants or agents cooperatively find good solutions to discrete optimization problems. The communication policy used by agents in ACO is to indirectly communicate through the environment by means of *stigmergy*. Stigmergy was first proposed by Grassé [6] to explain the construction of nests by the termites *Cubitermes* and *Macrotermes*. Grassé observed that when workers of *Macrotermes bellicosus* were placed in a container with some soil pellets, the insects carried about and put down pellets in an apparently random fashion after an exploration phase in which they moved through the container without taking any action. At this stage, a pellet just put down by a termite worker is often picked up and placed somewhere else by another worker. When a pellet is placed on top of another, the resultant structure appears to be much more attractive and termites soon start piling more pellets nearby, making the dropping spot even more attractive [17].

In ACO, we can see stigmergy in action whenever an artificial ant deposits a pheromone trail on a problem solution space. If an artificial ant come across a pheromone trail, it is attracted to it, very much like termites are attracted by clusters of soil pellets. By means of this indirect communication channel, ants share knowledge and the pheromone trail is a "blue print" to build a good solution to the problem at hand.

Another swarm intelligent system that relies on stigmergy, and that is perhaps more related to the behavior of termites, is ant-based clustering which was introduced by Deneubourg et al. [2] using a model for spatial sorting. A group of agents exhibiting the same behavior move randomly over a toroidal square grid. In the environment there are objects that were initially scattered in a random fashion. The objects can be picked up, moved or dropped in any free location on the grid. An object is picked up with high probability if it is not surrounded by other objects of the same type and is dropped by a loaded agent if its neighborhood is populated by other objects of the same type and the location of the agent has no object on it. Lumer and Faieta [12] generalized the spatial sorting algorithm to apply it to exploratory data analysis.

The implementations of the techniques described in the preceding paragraphs can be considered swarm intelligent systems. Both of them use indirect communication among agents through local modifications of the environment as their principal inter-agent communication policy. But, can we expect better results if we let agents communicate directly in ACO and ant-based clustering algorithms? Would the results obtained after doing so be problem dependent? Is it convenient to maintain the same communication policy during execution?

All these questions are related to the agents' communication policies and their effects in swarm intelligent systems. A communication policy represents the way an agent communicates with others. It must establish what information/knowledge to exchange, the way this exchange is to be done, and the appropriate moment to do so. A communication policy may be dynamic, that is, an agent may find convenient to change the way it communicates with other agents in a particular time or situation. It may also be selective, or in other words, it may apply only for a selected group of agents, etc. These and other properties may also be identified, and for all of them, there is a lack of knowledge regarding their effects on a particular problem when used in a swarm intelligent system.

By establishing well-founded guidelines for the design of communication policies among agents in swarm intelligent systems, we would be giving an important step towards the definition of a general methodology that would spread the practical use of swarm intelligence.

In this paper we focus on ant-based clustering algorithms, a particular kind of swarm intelligent system, and on the effects on the final clustering of letting agents use different communication policies.

3 Ant-Based Clustering Algorithms: A Case Study

In this section, we will describe a series of experiments that show how different inter-agent communication policies affect the performance and final outcome of a swarm intelligent system, in this case, an ant-based clustering algorithm. We will start by giving some background on ant-based clustering algorithms, then we will present our experiments setup and our results.

3.1 Background

Prior to the existence of ant-based clustering algorithms as such, Deneubourg et al. [2] proposed a computational model for spatial sorting. Deneubourg et al.'s model was later extended by Lumer and Faieta [12] to allow its application to exploratory data analysis. In their model, objects represent data items that belong to a database. These objects are randomly scattered on a periodic square grid on which randomly moving agents group them according to their similarity. In order to do that, a similarity (or dissimilarity) measure between pairs of data items is needed to compute the probabilities of picking and dropping data elements on the grid. In their model, the probability of picking a data element i is defined as $p_p(i) = \left(\frac{k_p}{k_p + f(i)}\right)^2$ where k_p is a constant and $f(i)$ is a similarity density measure with respect to element i. Likewise, the probability of dropping a data element is given by $p_d(i) = \begin{cases} 2f(i) & \text{if } f(i) < k_d \\ 1 & \text{otherwise} \end{cases}$ where k_d is a constant. The similarity density $f(i)$ for an element i, at a particular grid location τ, is defined as $f(i) = \max\left\{\frac{1}{s^2}\sum_{j\in Neigh(\tau)}\left(1-\frac{d(i,j)}{\alpha}\right), 0\right\}$ where s^2 is the size of the perception area $Neigh(\tau)$, centered at the location of the agent and α is a scaling factor of the dissimilarity measure $d(i,j)$ between elements i and j.

After the first appearance of this algorithm, many other variations of it have been proposed to improve its output quality [13,7], its convergence speed [8], and its applicability to large databases [16]. However, no previous works are known that study the effects of different communication strategies among agents in these algorithms.

3.2 Experiments Setup

In total, four different information exchange strategies were studied: direct information exchange for updating the agents' environment representations, direct information exchange for changing the agents' dropping spot search trajectories, intentional indirect information exchange for updating the agents' environment representations, and intentional indirect information exchange for changing the agents' dropping spot search trajectories.

To design the experiments, questions such as: What information will agents exchange? When will they exchange it? How will they do it? What will they do with that information? had to be answered. Although there are no general answers to them, we tried to explore four issues when we proposed answers to them. First, what are the effects on the performance of the algorithms when agents exchange information from different levels of abstraction? Second, what is the impact on the performance of the algorithms when agents exchange information in different ways? Third, what happens when agents use the information for different purposes? And fourth, what happens when agents choose to use immediately or after some delay the exchanged information?

Let us discuss how we coped with these issues and what were our results. The information that agents exchanged in the experiments belong to two different levels of abstraction: memorized grid locations on which an agent had dropped a data object, and pointers to promising dropping locations. The first choice was needed in order to compare the performance of the algorithms with and without communicating agents. In fact, this model is just a simple extension of Lumer an Faieta's short-term memory agents model. The second choice tries to implement the idea of map exchanging agents. Maps were implemented as growing neural gas networks or GNGs [5], which are distributed in the grid and in the attribute space of data objects. GNGs provide more information than just memorized dropping spots because they can adapt to changes in the spatial distribution of objects.

GNGs were designed to overcome some of the limitations of conventional self-organizing maps; namely, the *a priori* fixed number of neurons and the problem of "dead" neurons or neurons that do not update their weight vectors due to a misplacing in the input space. The GNG training algorithm successively adds new units to an initially small network by evaluating local statistical measures gathered during previous adaptation steps. In this way, a GNG network topology is generated incrementally by using a competitive Hebbian learning rule and its dimensionality depends on the input data and varies locally.

With regard to the moment in which agents should exchange information, we decided to couple this issue with the way agents were going to exchange information. That is, agents decided when to exchange information depending on the exchange method used. When agents exchanged information directly, they did it whenever they met on the grid. When they exchanged information indirectly, they did it whenever an agent came across an information packet on the grid.

One of the most critical part in the experiments design was to decide what agents should do with the exchanged information. Our hypothesis was that agents with information about the spatial distribution of data on the grid, would be able to choose the best location on which to drop an object (if they were loaded), or the best regions of the environment to explore (if they were unloaded). With informed decisions, agents could create better clusters in a faster way. We therefore decided to explore the idea by (i) letting agents represent their environment and after every exchange, update or enrich their representations, and (ii) changing their dropping spot search trajectories, i.e., they were allowed to "change their minds" regarding their supposed best dropping spot on the grid.

To evaluate the quality of the obtained clustering, the same validity measures used by Handl et al. [7] were used: the F-Measure, which gives us some idea of how well a clustering algorithm is identifying the classes present in a database using the information of the correct classification[1]; the Rand statistic, which is a similarity measure between the known perfect classification C and the partition generated by the clustering algorithm P, considering all pairwise assignments; the Dunn index, which measures how compact and well separated are the identified clusters; and the intra-cluster variance, which measures how similar are the elements belonging to the same cluster.

We used two real data collections from the UCI Machine Learning Repository [9]. These were: the iris plant and wine recognition databases. To eliminate the bias on similarity measures provoked by different scales within data attributes, both databases were standardized. The similarity measure used in all the experiments was the cosine metric[2]. The agents' picking and dropping probabilities were computed using the same expressions as Lumer and Faieta. However, since the cosine metric is a similarity measure, the expression used to compute the similarity density is not directly applicable. Therefore, for the local similarity density $f(i)$, we used $f(i) = \frac{1}{s^2} \sum_{j \in Neigh(\tau)} \left(\frac{1}{1 + e^{-S\frac{d(i,j)}{\alpha} + D}} \right)$ where S is the steepness of the response curve and D serves as a displacement factor. In our experiments, S was fixed to 5 because it provides a similarity value close to 0 when the cosine measure is minimum, that is, when the cosine measure gives a

[1] Which is available for our experiments.

[2] In preliminary experiments, it proved to give better results than Euclidean distance.

value of -1, and D to 1 because this allows us to better distinguish vectors with separation angles between 0 and $\pi/2$.

To observe the effects of information exchange among agents during the clustering process, the data partition obtained every 10,000 simulation cycles was evaluated using the validity measures described above. Each simulation cycle was composed of N individual actions, where N was the number of agents in the simulation. All algorithms were tested 30 times with every database for 1,000,000 simulation cycles. We tried with populations of 10 and 30 agents within an environment of 100×100 locations in all the experiments.

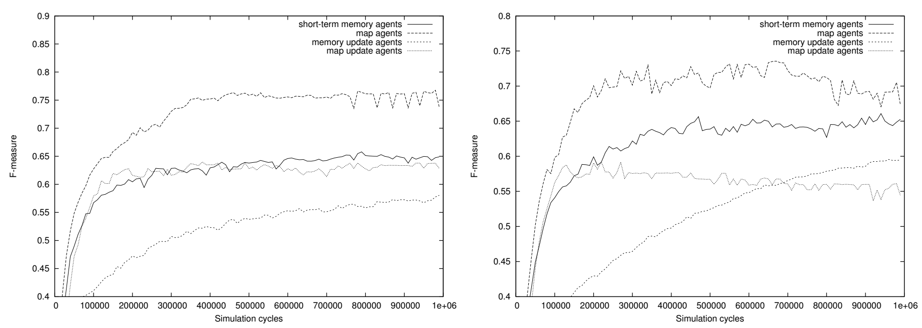
4 Results

The following sections present in detail the conclusions drawn from the experimental results with each of the tested strategies. For space restrictions, we refer the interested reader to [15, 14] for the complete set of results. In this paper we will only show some selected graphs to support our conclusions.

4.1 Direct Information Exchange

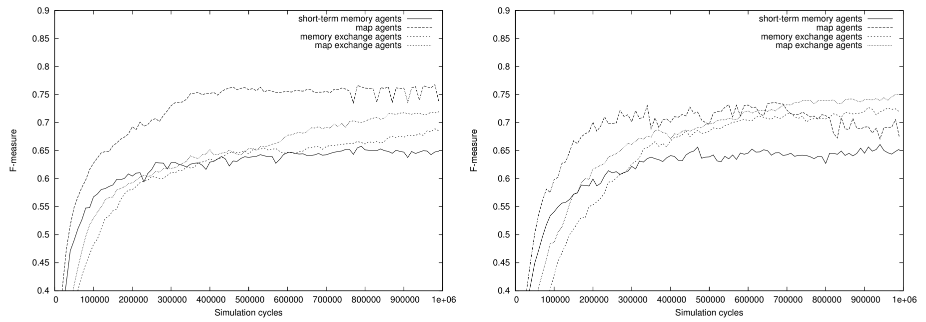
As we said before, direct information exchange occurs only when two or more agents meet at a location on the grid. Hence, the probability of an encounter between two agents moving randomly raises as the number of agents is increased (assuming a constant size of the grid). In these experiments, we tried to take advantage of this fact and use it to study the effect of increasing the information exchange frequency among agents. This is the reason of using two different sizes of agent populations in all the experiments.

The results obtained when the exchanged information was used for updating the agents' environment representations are somewhat discouraging. The worst performing algorithm is the one with map updating agents, and the second worst



(a) Iris database (b) Wine database

Fig. 1. F-Measure scores over time for the Iris Plant and Wine databases of all tested algorithms. The results were obtained using 10 agents. Values closer to 1 are better.



(a) Iris database (b) Wine database

Fig. 2. F-Measure scores over time for the Iris Plant and Wine databases of all tested algorithms. The results were obtained using 10 agents. Values closer to 1 are better.

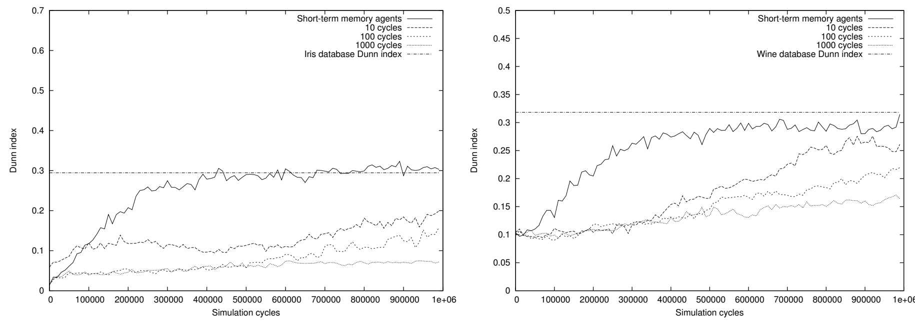
is the memory updating agents algorithm. In the case of the memory updating agents algorithm, the relatively poor performance is not as discouraging as is the obtained with map creating agents. Figure 1 shows the F-Measure scores obtained by the tested algorithms on both test databases using 10 agents.

The results obtained when the exchanged information was used for changing the agents' dropping spot search trajectory are quite different from the previous ones. In this case, the clustering quality is improved by the communicating agents algorithms. Figure 2 shows the F-Measure scores obtained by the tested algorithms on both test databases using 10 agents.

4.2 Indirect Information Exchange

In the experiments run for exploring the effects of intentional indirect information exchange among agents in ant-based clustering algorithms, agents lay packets which contain information about data distribution on the environment for others to pick and use. This strategy is inspired by the anal trophallaxis phenomenon [17] among social insects but it also has other reasons. Direct communication among agents in ant-based clustering has two disadvantages: (i) even when the number of exchanges increases, we cannot expect many of them to happen since the number of agents must be kept small (for performance reasons), and (ii) many exchanges do not have any effect since agents walk in a randomly fashion, i.e., two agents coincide many times, over and over again, before they follow different trajectories. So the idea is that if we let agents lay information on their environment, it could be possible to increase dramatically the number of exchanges without even increasing the number of agents.

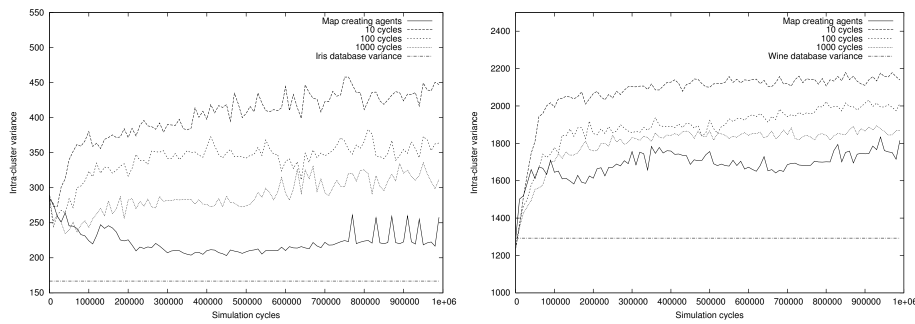
Two information laying policies were studied: a periodic laying policy and an adaptive laying policy. With the periodic laying policy, an agent drops information packets every given number of simulation cycles. With the adaptive laying policy, an agent drops information after it has modified the environment and a given number of simulation cycles have passed.



(a) Iris database (b) Wine database

Fig. 3. Dunn index scores over time for the Iris Plant and Wine databases of all tested algorithms. The results were obtained using 30 agents using a periodic laying policy. As a reference, the Dunn index of the correct clustering is shown.

The results obtained when the exchanged information was used for updating the agents' environment representations, show that the more information available to the agents, the better the performance. With a periodic laying frequency, the higher the frequency, the better. And with the adaptive laying policy, the shorter the delay, the better. This results confirm the intuition which says that to maintain an up-to-date environment representation, an agent has to acquire fresh information all the time. Figure 3 shows the Dunn index scores obtained by the tested algorithms on both test databases using 30 agents and a periodic laying policy.



(a) Iris database (b) Wine database

Fig. 4. Total intra-cluster variance scores over time for the Iris Plant and Wine databases of all tested algorithms. The results were obtained using 10 agents using a periodic laying policy. As a reference, the total intra-cluster variance of the correct clustering is shown.

When agents use the information only to decide whether to change their dropping spot search trajectory, the more information available to them, the worse. This result is also sound if we think of agents as "changing their minds" based on the information provided by other agents. An agent may receive contradicting information or may be misleaded to a nonpromising region in the environment. Therefore, with this information exploitation strategy, high laying frequencies and short delays have negative impact on the algorithms performance. Figure 4 shows the Total intra-cluster variance scores obtained by the tested algorithms on both test databases using 10 agents and a periodic laying policy.

5 Conclusions

In this paper, the effects on clustering quality and convergence speed of direct and indirect communication among agents in ant-based clustering algorithms were studied. The results show that nonstigmergic communication among agents in these algorithms has effects on the final clustering obtained by the algorithm. The final effects depend on the type of information exchanged, its use, its availability and the number of participating agents. Our results confirm that different communication policies in swarm intelligent systems have effects on their performance. This is why we need to formalize the effects of letting agents use different communication policies. This is a first step towards that goal. With this knowledge and considering the characteristics of a particular problem, we could either improve the performance or permit the application of a swarm intelligent system to solve it. Future work should be focused on studying the effects of using different communication policies in ACO and other swarm-based approaches.

Acknowledgments

This work was supported in part by the ITESM research chair on distributed knowledge and intelligent agents technologies, CAT-011.

References

1. E. Bonabeau, M. Dorigo, and G. Theralauz. *Swarm Intelligence. From Natural to Artificial Systems.* Oxford University Press, 1999.
2. J.-L. Deneubourg, S.Goss, N Franks, A. Sendova-Franks, C. Detrain, and L. Chretien. The dynamics of collective sorting: Robot-like ants and ant-like robots. In *Proceedings of the First International Conference on Simulation of Adaptive Behavior: From Animals to Animats*, pages 356–365. MIT Press, 1991.
3. M. Dorigo, G. Di Caro, and L. M. Gambardella. Ant algorithms for discrete optimization. *Artificial Life*, 5(2):137–172, 1999.
4. M. Dorigo and T. Stützle. *Ant Colony Optimization.* MIT Press, Cambridge, MA, 2004.
5. B. Fritzke. A growing neural gas network learns topologies. In *Advances in Neural Information Processing Systems 7.* MIT Press, 1995.

6. P.-P. Grassé. La reconstruction du nid et les coordinations inter-individuelles chez bellicositermes natalensis et cubitermes sp. la theorie de la stigmergie: Essai d'interpretation des termites constructeurs. *Insectes Sociaux*, 6(1):41–83, 1959.
7. J. Handl, J. Knowles, and M. Dorigo. Ant-based clustering and topographic mapping. *Artificial Life*, 11(2), 2005. In press.
8. J. Handl and B. Meyer. Improved ant-based clustering and sorting in a document retrieval interface. In *Proceedings of the Seventh International Conference on Parallel Problem Solving from Nature (PPSN VII)*, volume LNCS 2439, pages 913–923. Springer-Verlag, 2002.
9. S. Hettich, C.L. Blake, and C.J. Merz. UCI repository of machine learning databases [http://www.ics.uci.edu/∼mlearn/mlrepository.html], 1998.
10. O. Holland and C. Melhuish. Stigmergy, self-organization, and sorting in collective robotics. *Artificial Life*, 5:173–202, 1999.
11. J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks*, pages 1942–1948, 1995.
12. E. D. Lumer and B. Faieta. Diversity and adaptation in populations of clustering ants. In *Proceedings of the Third International Conference on Simulation of Adaptive Behavior: From Animals to Animats 3*, pages 501–508. MIT Press, 1994.
13. N. Monmarché, M. Slimane, and G. Venturini. On improving clustering in numerical databases with artificial ants. In D. Floreano, J.D. Nicoud, and F. Mondala, editors, *5th European Conference on Artificial Life (ECAL'99), Lecture Notes in Artificial Intelligence*, volume 1674, pages 626–635. Springer-Verlag, 1999.
14. M. A. Montes de Oca. Effects on clustering quality of direct and indirect communication among agents in ant-based clustering algorithms. Master's thesis, Instituto Tecnológico y de Estudios Superiores de Monterrey. Campus Monterrey, 2005.
15. M. A. Montes de Oca, L. Garrido, and J. L. Aguirre. Efectos de la comunicación directa entre agentes en los algoritmos de agrupación de clases basados en el comportamiento de insectos sociales. *Inteligencia Artificial*, 9(25):59–69, 2005.
16. M. A. Montes de Oca, L. Garrido, and J. L. Aguirre. An hybridization of an ant-based clustering algorithm with growing neural gas networks for classification tasks. In *Proceedings of the 2005 ACM Symposium on Applied Computing*, pages 9–13. ACM, 2005.
17. E. O. Wilson. *The Insect Societies*. The Belkap Press of Harvard University Press, 1971.