Sharing ressources with artificial ants

Christophe Guéret[1], Nicolas Monmarché[1], Mohamed Slimane[1]
[1]Université François Rabelais de Tours, Laboratoire d'Informatique
64 Avenue Jean Portalis, 37200 Tours - France
{christophe.gueret,nicolas.monmarche,mohamed.slimane}@univ-tours.fr

Abstract

As networks are growing up , more and more information becomes available every day. Despite the presence of software enabling communications and content sharing, they are not always shared among people inside networks. We present here an architecture aimed at helping people to share informations and find collaborators inside an organization. It is part of our PIAF framework, an intelligent agent system we use to develop recommender and personalization software. The main contribution of this paper is the introduction of principles of stigmergy and artificial ants to model data flows in a social network.

1 Introduction

While browsing the web, communicate with friends, read books, *etc*: everybody is used to searching, fetching and storing a huge amount of data. Despite the existance of many ways to manage and share data, communication between data providers and data scouts is still a problem, particularly in large organizations. Also, scout and provider can be the same person and lead to the same difficulties: who has never experimented having lost a file or a contact in the thousand of MB of his/her hard drive ? While the data is on the disk (*ie* provided by the user), the scout (the same user) is unable to find it. In larger organizations involving many people interested in many different subjects, the crucial question is "who is worth communicating with?". As people and topics are constantly evolving, a global knowledge of every colleagues' center of interest is quite impossible to maintain, and so finding someone able to provide help is a tricky task. When an organization grows, the probability that two people could be interested in the same subject and do similar tasks increases. In the last decades, collaborative tools have been proposed to help people sharing their web browsing results [13, 14, 19] and files [11, 1, 2]. Also, tools have been proposed to help finding potential collaborators within the network [3, 8, 16].

A recurring phenomenon as been frequently observed in large networks of people: people tends to congregate forming an ensemble of small sets in the global network [11]. This is usually referenced as the "small world" effect [20], characterized by dense groups loosely connected. This small world can also be considered as a particular kind of social network; that is a network of users connected according to relationships [9]. Social networks are similar to peer-to-peer networks where each host can act both as a client and a server for all possible types of exchange. Peer-to-peer networks are particularly suited to cope with dynamic network topology and collaborative systems where every user may be looking for information or providing some.

A complex adaptive system aimed at helping people to share informations has to cope with some difficulties:

- Lack of mutual awareness: sharing content efficiently implies a global knowledge. In order not to bother every single one, every user should know what his/her peers are interested in. The problem is users may randomly appear and disappear in the network. Also they may be interested in different domains or make spuradious searches from time to time.

- Lack of motivation from users in using the system: sharing content may involve, for instance, sending emails to people inside the network or using a dedicated tool to tell them about what they have found. Both examples may change user's habits and require them to make an effort. Users do not like to change their habits and such solutions may break their motivation and dissuade them from using the software.

- Users can not define precisely what they are interested in: if we take the example of web browsing,

1-4244-0054-6/06/$20.00

users are most likely to jump from page to page as they find interesting links rather than following a precise and predeterminated path.

The main contribution of this paper consists in using stigmergy principles to model communications and promote collaborators' discovery inside social networks. Considering a population of entities in their environment, stigmergy [10] is observed when local environmental changes performed by an individual have an influence on other's behaviour. Ants provide a good example of a stimergetic system: thanks to chemical substances called pheromones, they are able to mark their path and then share this knowledge with co-workers.

The remaining of this paper is constituted as follows. Next section will present communications observed in a P2P network and common solutions found to handle them. Section 3 presents how those communications could be improved using a model based on artificial ants called PIAF. A discussion about this project and other existing systems is in section 4. We finally conclude on this work and perspectives in section 5.

2 Communications inside social networks

Social networks rhyme with communications. In last decades, many solutions have been proposed to help people sharing resources and find collaborators inside a network. Some of them will be discussed here, starting by solutions to exchange data and knowledge followed by techniques enabling matchmaking process in the network.

2.1 Sharing data and knowledge

Looking for a particular data in P2P networks is a tricky task. Finding a particular data X in the network is a "lookup" operation aimed at answering to "where can I find X ?". The expected answer is a list of peers sharing X. While in a client/server architecture, the server is the only data warehouse, in a P2P network every peers serves it's own content: each peer knows what he shares and saves this information as an index. A look up operation has to browse through all those indexes to compute an answer. Indexes may be stored locally [1] or, to speed up lookup algorithms, be centralized [2] or distributed [12, 17]. Whatever the solution is, using an index of filenames may be only used to query files by their name but not by their content. Sharing files by their content would involve creating a file named after each keyword and share it. Cuenca *et al.* [6] have proposed to share a reverse index of keywords in documents as a smarter solution. The algorithm proposed by Zhu and *et al.* [21] allows a user to search for similar files in the network. Files shared are grouped in clusters and their index based on files names is replaced by indexes reflecting their similarity.

Sharing knowledge is not easier. If a peer wants to have all the network members discovering a great web site he has found, he will not have any file to share but only the site's location. The same goes where the question is to inform about a great book or date for a manifestation. Emails, usenet newsgroups, forums and more recently social portals such as Orkut (`www.orkut.com`) are to enable people to share knowledge. Those tools are usefull but enforce users to:

- Explicitly decide to share information and decide whom he will share it with. Considering email, some potentially interested people may be forgotten in the receivers' list. Spamming comes has an example of the opposite case.

- Monitor many source of informations. Most Internet users have two or more mail boxes, have subscribed to many forums and pay a visit to newsgroups or community portals from time to time.

The common point of look up algorithms and knowledge sharing tools is the system of query/answer: directly or not, users have to precisely know what they are looking for and ask for it. Directly if a query is requested and indirectly if they, for instance, have to register to a forum. Users habits do not always rely on this scheme: they may be interested by files shared in the network although they are not looking for it. Web browsing usualy consists in jumping from site to site as an apparently interesting link is found. Doing so, a user may finally discover and download a file he was not trying to find. Also, since query formulation involves selecting a set of words, users may forget to include some key words and not formulate the right question when querying the network.

2.2 Finding collaborators

Each peer is connected to a subset of all peers in the network. According to small world principles, collaborators should be grouped together to optimize communications in the whole network. Finding whom he wants to be connected with is a task belonging to the peer: according to his needs he has to find which peers inside the networks may help him and connect them.

3 Problem and related work

The problem to solve is formulated by two goals:

- Dispatch accross the network a set of informations in order for each peer to gain access to informations they may find interesting

- Find and group collaborators inside the network by dropping useless connections and establishing new ones as necessary

We have choosen to use a model based on artificial ant colonies. Such models have been already used to solve a large variety of load balancing and routing problems inside a network. We will discuss some of them and spot differences with the solution introduced hereby.

In a foresight work by Dorigo [7], ants have first been used to find the optimal route inside a network, that is the minimal distance between two nodes of a graph. Initialy, a set of m ants are randomly located on nodes. Iteratively, all ants move from the node they are to an other according to the edge's length and a quantity of pheromones stored in a routing table. Once all ants have moved, node's routing tables are updated with the path they found.

Bertelle *et al.* [5] have proposed a model inspired from Dorigo and dedicated to perform load-balancing. Considering nodes as being the set of tasks for a distributed application goals are to (1) equilibrate load on each machine and (2) minimize communications between two machines. Edges are weighted according to the amount of communication needed by the concerned tasks. The algorithm tends to group on a same computer ressources tasks having highest edge's weights. At the opposite to this algorithm, in [4, 15] the network represents connected computing ressources. Ants transport tasks within the network following a simple strategy: continuously take a task from an overloaded node and walk across the network until an underloaded node is found. Then, assign the task to it and move in the network until an other overloaded node is found.

More related to our problem is the algorithm proposed by Schmitz [18]. The solution is based on routing tables on each node and relies on message forwarding to discover new peers. Ants walk from node to node to fetch and bring back expertises data from peers they cross by. Once they are back to their nest, those expertises are used to decided which peers the nest should be connected to. Walks are performed following a greedy strategy of electing the destination which is more likely to be interesting: connection to a peer whose expertise is close to the expertise of the peer ant cames from. A limitation of this algorithm is that, when asking a peer his expertise, an ant ignores it's neighboorhood. That is, the data fetched is only representative of the peer and contains no information related to peer around him.

In order to find groups, it may be usefull to have a global notion of expertise: instead of doing rewiring process based on information likes "peer X is interested in Y" it could be done with something like "in the neighboorhood of X, peers are interested by Y". We have defined and evaluated routing and rewiring algorithms based on this idea.

4 Basics and definitions

4.1 Model for network and peers

The P2P network can be seen as a directed graph $G(V, E)$ with a set of nodes V and a set of edges $E \subseteq V \times V$. Considering two nodes u and v, an edge $e = (u, v)$ designs the connection from u to v. PIAF framework is running on each node. This framework is made of components separated into three layers (see figure 1): (1) Low level components are to deal with storage and communications tasks. (2) The managers in the middleware are in charge of performing network's rewiring tasks and informations management. (3) Agents then supply an interface between the user and the network of informations.

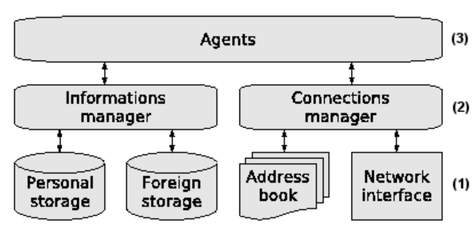

Figure 1. PIAF framework architecture

4.2 Model for informations

We define one "information" as a meta-data defined on Ω representative of any ressource shared in the network. A similarity mesure $sim : \Omega \times \Omega \mapsto [0, 1]$ is defined to determine the similarity between elements of Ω.

Schmitz [18] considers Ω as being an ontology. In few words, an ontology is a semantic structure wich defines t-uples between a set of *concepts* and a set of *relations*. For instance, usings the concepts of $Professor$ and $PhDStudent$, a t-uple $supervise(Professor, PhDStudent)$ defines the relation of *supervise* between them. Since concepts are partially ordered into *subconcepts* (hence producing a structure of graph) the number of edges between two concepts can be used to define their similarity.

In our experimentations, Ω is defined as a Vector Space Model (VSM). In this representation, an information is a vector $X = x_i$ and the similarity can be expressed as a cosinus equation:

$$sim(X,Y) = \frac{\prod_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}} \quad (1)$$

The advantage of this representation is to get rid of ontologies' limitations but, as a drawback, this model produces "bag of words". That is, an unordered list of words. Hence, considering such a vector, the semantic induced by the order of words in the sentence is lost.

4.3 Model for users

Informations stored by a given peer may have either been produced by the peer itself or fetched from the network. In the first case, they are stored in the "personal storage" while in the second case they fall into the "foreign storage". The content of those storages are respectively designed as S_p and S_f. Informations present in the personal storage are supposed to be representative of user's interests. Those interests are represented as an information $W_u^{(t)}$ defined as the mean vector of the content of $S_p^{(t)}$.

$$W_u^{(t)} = \frac{1}{|S_p^{(t)}|} \sum_{X \in S_p^{(t)}} X \quad (2)$$

4.4 Model for ants and pheromones

Inside each nest, some ants are in charge of copying informations to other nests. According to different strategies presented in the next section, this activity generates a diffusion of informations in the network. While they are going from one nest to one other bringing with them an information, ants will lay down pheromones.

Let u and v be two peers in the network, linked by the connection $e(u, v)$. The vector of pheromones on this connection is denoted as $\tau(u, v)$.

$$\tau^{(t)}(u,v) = (1-\alpha)\tau^{(t-1)}(u,v) + \alpha X \quad (3)$$

with α the amount of pheromones to be added. The value of α is designed to be proportional to traffic on the road. The more informations are exchanged through a road, the more pheromones quantity will be important on it.

$$\alpha = \rho e^{-\frac{(t - l(u,v))^2}{\sigma}} \quad (4)$$

with ρ and σ two coefficient used to respectively temperate the impact of α in the formulation of τ and to adjust it's value in time. $l(u, v)$ is a timestamp of last time an information was transfered through the connection $e = (u, v)$

5 Diffusion of informations

Informations are data blocks exchanged between peers. From an artificial ant point of view, they are food needed to be sent to other nests. We define two kinds of ants to handle the transfers. Both are using the sames behavioral rules: (1) Transfer: the ant transfers a duplicate of an information from one node to an other and (2) Return: the ant returns back to his nest. While in a "Transfer" state, an ant takes an information from one of the storages and duplicates it. Then, it goes to an other nest to store this duplicate into peer's foreign storage. Once this is done, it changes it's state to be in "Return" state and go back to the nest it cames from.

The choice of the storage zone to take an information from depends of the kind of ant considered. A "personal ant" will use the personal storage as a source for information whereas a "foreign ant" will serve itself in the foreign storage.

5.1 Diffusion ant for personal informations

Those ants are designed to diffuse information produced locally by the peer. On moving an information X, ants have to choose between connected peers the one which is more likely to be interested. This decision is based on a comparison between X and pheromones present on the roads to those peers.

Let us consider an item X randomly taken from the personal storage and the neigboorhood $\Gamma_u^{(t)} = \{v \in V \backslash \{u\} \mid (u, v) \in E^{(t)}\}$ of u. During its decision process, the ant starts by computing the similarity between X and pheromones present on each road. Given this first data, neighboors are sorted in two groups: "interested" (I) and "non interested" (J)

- The "interested" group is composed by all the peers having a similarity different from 0

$$I = \{\ v \mid sim(u, \tau^{(t)}(u,v)) \neq 0,\ v \in \Gamma_u^{(t)}\} \quad (5)$$

- "non interested" group is made of all remaining peers. That is whose similarity is equal to zero.

$$J = \{\ v \mid v \in \Gamma_u^{(t)},\ v \notin I\} \tag{6}$$

Inside each group, a connection to a peer v_i have a probability $q^{(t)}(v_i)$ of being chosen, according to their relative interest:

- Within the "non interested" group, peers may be equiproportionally chosen

$$\forall v_i \in J, \quad q^{(t)}(v_i) = \frac{1}{|J|} \tag{7}$$

- A peer member of the "interested" group have a probability proportional to its relative similarity compared to other peers

$$\forall v_i \in I, \quad q^{(t)}(v_i) = \frac{sim(X, \tau^{(t)}(u, v_i))}{\sum_{v_j \in I} sim(X, \tau^{(t)}(u, v_j))} \tag{8}$$

An optimum data flow would consist in always sending informations only to most interested peers. On the other hand, to find new interested peers to connect to, network exploration involves trying to send informations to neighbors even if they seem not to be interested. Therefore, a trade-off must be found between optimized transfers and network exploration. The solution proposed here consists in giving ant agents a notion of freewill. According to a probability f, an ant chooses to send an information either to a peer in the group I or J. The lower the value of f is, the less network exploration would be done. When an empty group is elected, the ant "decides" not to move.

Let's denote $c = \{I, J\}$ the ant choice with $p(c = I) = f$ and $p(c = J) = 1 - f$. A connection $e = (u, v_i)$ may be choosen according to $p^{(t)}(u, v_i)$ defined as:

$$p^{(t)}(u, v_i) = \begin{cases} \delta_{cJ} q^{(t)}(v_i) & \text{if} \quad e \in J \\ \delta_{cI} q^{(t)}(v_i) & \text{if} \quad e \in I \end{cases} \tag{9}$$

with δ_{cI} defined as $\delta_{cI} = 1$ if $c = I$ and 0 if $c \neq I$. The same goes for δ_{cJ}

5.2 Diffusion ant for foreign informations

Since the ant previously defined only takes informations from the personal storage, an other kind of ant takes care of foreign informations. In its general behavior it acts like a personal ant: it takes an information and send it to a peer. But informations stored in the foreign zone are not significative of peer's centers of interest. Relaying them on the network may distort values of the interests as computed in equation 5. Nonetheless, it is necessary to forward informations to other peers to diffuse informations not locally produced.

So, to allow such relaying, we slightly adjust the equation 9 to introduce a condition: the ant moves an informations from the foreign storage only if it founds it interesting. Let's Q being this condition:

$$Q = \begin{cases} 1 & \text{if} \quad sim(W_u^{(t)}, X) > \lambda \\ 0 & \text{if} \quad sim(W_u^{(t)}, X) \leq \lambda \end{cases} \tag{10}$$

with $\lambda \in [0, 1]$ the minimum similarity required to find the information X interesting. The equation 9 becomes:

$$p^{(t)}(u, v_i) = \begin{cases} \delta_{cJ} Q q^{(t)}(v_i) & \text{if} \quad e \in J \\ \delta_{cI} Q q^{(t)}(v_i) & \text{if} \quad e \in I \end{cases} \tag{11}$$

It can be noted that this ant behavior is similar to the interaction a user would have with the system. Actually, a user U may periodically have a look into the foreign storage to see if an information relevant to him has appeared in it. Supposing he founds an information X created by U' and access it, a new information X' is generated and stored in the personal storage. This information X' is similar to X and will be diffused in the network. Only the name of the creator differenciate X from X'.

6 Rewiring strategies

6.1 Satisfaction

Rewiring process requires from a peer to be able to estimate the quality of his connections at a given time t. This quality may be qualitative or quantitative depending of the parameter to optimize. To have a network able to quickly react to changes, the quantity of data exchanged during a given time will be of interest whereas if efficiency is the priority, the similarity between user's needs and informations fetched will be spotted. We define this quality as

$$s^{(t)}(u, v) = sim(W_u^{(t)}, W_v^{*(t)}) \tag{12}$$

with $W_v^{*(t)}$ the expertise vector of peer v as seen by u. We introduce two different strategies to compute $W_v^{*(t)}$: u can have v sending him this vector. u can guess it from the pheromones on the connection from v to u. This lead to the definition of two values for the satisfaction s respectively denoted as s_r and s_e were r and e stands for **r**eal **e**stimated.

6.1.1 Asking for peer's expertises

Asking a peer for his expertise is the first way to instanciate W_v^*. To do so, u sends a message to v and given the answer computes s_r:

$$s_r^{(t)}(u,v) = sim(W_u^{(t)}, W_v^{(t)}) \tag{13}$$

To reduce network overload generated by those messages, this information can be cached and updated at a different frequency than the one used to check satisfaction.

Those two strategies are somehow similar to algorithms found in [7], [5] and [4] were messages or agents are sent over the network to collect and bring back a set of expertises.

6.1.2 Estimate peer's expertises

The other way is to estimate trails laid by informations on the connection $e = (u,v)$.

$$s_e^{(t)}(u,v) = sim(W_u^{(t)}, \tau^{(t)}(u,v)) \tag{14}$$

Using this definition provides two advantages: First, network speed between u and v is implicitly taken into account by the evaporation of pheromones and their influence on values of $\tau(u,v)$. Second, since $\tau^{(t)}(u,v)$ is computed as a linear combinaison of informations transiting on $e = (u,v)$, this estimated expertise may contains informations concerning neighbours of v.

6.2 Rewiring

Rewiring processes is considered appart from informations flows. To optimize its neighborhood, i.e. being connected only with peers whose interests are close to his, a peer u will periodically initiate a checkup algorithm. This is performed by the connection manager and based on the following rules:

1. If the average of satisfaction $s^{(t)}(u,v)$ for each connection (u,v) is beyond a ceil λ try to connect to a new peer fetched from the addressbook.

2. In case there is no place left for a new connection, drop the worst connection before establish the new one.

3. In case all peers in the addressbook are already connected, ask for the best connected peer to make a suggestion.

Those rules are the bases for the algoritm **??**. The `askForSuggestion` procedure is used to discover new peers in the network. For a given peer u, it consists in sending to an other peer v his expertise $W_u^{(t)}$. Peer v answers to u by telling him the name of the peer having the best similarity with $W_u^{(t)}$, among everyone listed in his own addressbook.

7 Graph clustering

In a group, people tends to surrogate with other people sharing the same interests. We will introduce here three coefficients used to verify it in the network starting by the original definition by Watts [20]. To illustrate the impact of differences, each definition will be followed by an exemple on the network given figure 2.

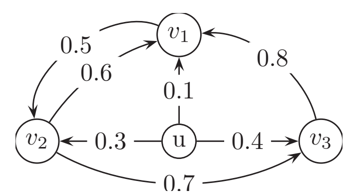

Figure 2. Exemple of graph

This exemple is made of 4 peers: u, v_1, v_2 and v_3, values on arcs indicate their similarity at an instant t. Considering u, $\Gamma_u^{(t)} = \{v_1, v_2, v_3\}$ and $k_u^{(t)} = |\Gamma_u^{(t)}| = 3$. As seen on the graph, u is in a cluster of peers almost similar to each other ($s \geq 0.5$) but u itself isn't very similar to his neighbors ($s < 0.5$).

7.1 Clustering coefficient

Considering the neighboorhood Γ_u of a peer u, the clustering coefficient has been defined as the number of existing connections within Γ_u divided by the number of possible connections:

$$\gamma_u^{(t)} = \frac{\mid E^{(t)}(\Gamma_u^{(t)}) \mid}{k_u^{(t)}(k_u^{(t)} - 1)} \tag{15}$$

with $E^{(t)}(\Gamma_u^{(t)}) = \{(u', v') \in E^{(t)} \mid u' \in \Gamma_u^{(t)}, v' \in \Gamma_u^{(t)}\}$. This value indicates how densely connected the neighborhood is. If it is equal to 1, every set of two peers in $\Gamma_u^{(t)}$ are connected to each other and so, peer u is connected to a very dense group. Applied to the example $\gamma_u^{(t)} = \frac{|\{(v_1,v_2),(v_2,v_1),(v_2,v_3),(v_3,v_1)\}|}{3*2} = 0.66$

7.2 Weighted clustering coefficient

In its original definition, the clustering coefficient only takes into account the notion of connection being present or not. Schmitz [18] has extended this definition to include the similarity the peer has with its neighbors:

$$\gamma_u^{w(t)} = \frac{\sum_{r\in\Gamma_u^{(t)}} s^{(t)}(u,r)|\{z\in\Gamma_u^{(t)} : (r,z)\in E^{(t)}\}|}{k_u^{(t)}(k_u^{(t)}-1)} \quad (16)$$

Each edge of a neighboor w counts only as much as the similarity between u and w. Applied to the example, this equation gives $\gamma_u^{w(t)} = \frac{0.1*1+0.4*1+0.3*2}{6} = 0.18$

7.3 Full weighted clustering coefficient

We go further in taking into account the similarity by adding the values of s between neighbors.

$$\gamma_u^{W(t)} = \frac{\sum_{r\in\Gamma_u^{(t)}} s^{(t)}(u,r)\sum_{z\in\Gamma_u^{(t)}:(r,z)\in E^{(t)}} s^{(t)}(r,z)}{k_u(k_u-1)} \quad (17)$$

Compared to the previous formulation, this equation also integrates an idea of neighboorhood heterogeneity. Applied to the example, this equation gives the lowest value $\gamma_u^{W(t)} = \frac{0.1*0.5+0.4*0.8+0.3*(0.6+0.7)}{6} = 0.13$

8 Evaluation

The system is implemented with the OMNETPP discrete event simulation system (`www.omnetpp.org`). We have defined a network of 80 peers interested in 4 different themes. On each peer, 4 agents are used: one ant of each type as introduced in former sections and 2 bots. A bot is an automated agent used to simulate the presence of a user. The first bot as in charge the task of creating new informations and the second periodically look into the foreign storage to see if there is something interesting for him. In every test, two strategies to have a peer's expertise were compared: ask for it or estimate it.

The goal is to compare the efficency of data gathered from the traffic between peers to data fetched by direct communications.

8.1 Satisfaction

The first results to be studied is the average of satisfaction for all the peers in the network. Results comparing results of using s_e and s_r are shown in figure 3.

During the first fraction of time, strategy choice does not make a difference. This is due to the time required to the pheromones to become effective and have an impact on communications. Later on, estimating peer's expertise appear to be a more effective strategy.

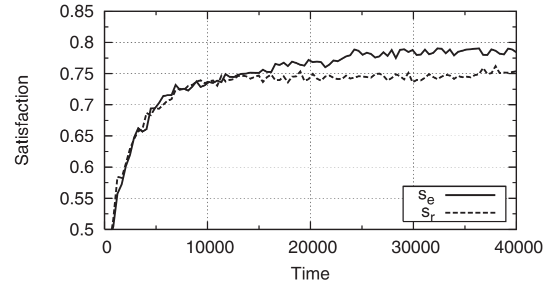

Figure 3. Evolution of peer's satisfactions

8.2 Clustering coefficients

Clustering coefficients have been tested for the three equations 15, 16 and 17 previously defined. The corresponding results are respectively shown on figures 4, 5 and 6.

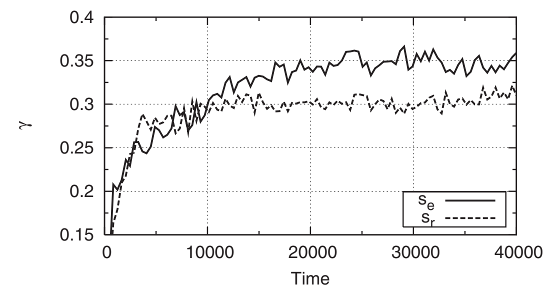

Figure 4. Evolution of clustering coefficient

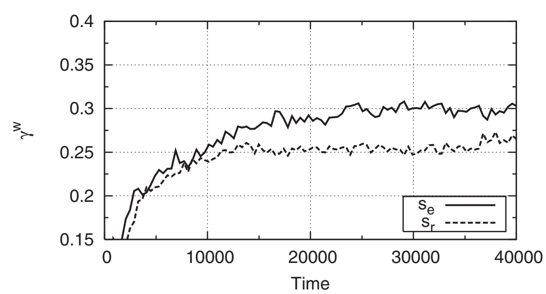

Figure 5. Evolution of weighted clustering coefficient

In all considerations, using estimated expertises is the most effective strategy. Also, the limit for the estimated value for the weighted clustering coefficient (figure 5) is equal to the limit of clustering coefficient with an asked value (figure 4). Which tends to prove that better groups are formed.

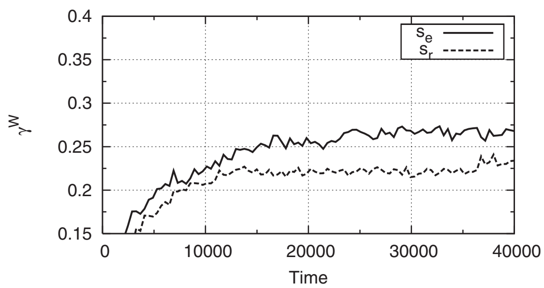

Figure 6. Evolution of fully weighted clustering coefficient

9 Conclusion

We have presented in this paper a new kind of data sharing algorithm for social networks. Using principles of stigmergy and artificial ant model, it allows the design of a non intrusive and dynamic framework to build communication software. Future directions for this work are the implementation of sensors for different applications and "real life" tests for the data diffusion algorithm. In particular, tests with real users instead of bots would allow a better estimation of parameters for ants.

References

[1] Clip2: Gnutella protocol specifications. `http://www.clip2.com`.

[2] Napster file sharing. `http://www.napster.com`.

[3] C. Aguirre, J. M.-M. noz, and R. Huerta. A low cost/high performance scalable topology for multi-agent collaboration. In *Autonomous Agents 2000 Workshop, Agents in Industry*, 2000.

[4] O. Babaoglu, H. Meling, and A. Montresor. Anthill: A framework for the development of agent-based peer-to-peer systems. In *Proceedings of the 22th International Conference on Distributed Computing Systems (ICDCS '02)*, Vienna, Austria, July 2002.

[5] C. Bertelle, A. Dutot, F. Guinand, and D. Olivier. Simulations distribues par un algorithme fourmi. In *RENPAR'15/CFSE'3/SympAAA'2003*, La Colle sur Loup, France, Octobre 2003.

[6] F. M. Cuenca-acuna and T. D. Nguyen. Text-based content search and retrieval in ad hoc p2p communities. In *International Workshop on Peer-to-Peer Computing*, May 19-24 2002.

[7] M. Dorigo, V. Maniezzo, and A. Colorni. The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, 26(1):29–41, 1996.

[8] N. Eagle and A. Pentland. Collaboration from conversation. In *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW'02)*, New Orleans, Louisiana, November 2002.

[9] N. Eagle and A. Pentland. Social network computing. In *Submitted to Fifth Conference on Ubiquitous Computing (UbiComp)*, Seattle, October 2003.

[10] P. Grassé. La reconstruction du nid et les coordinations inter-individuelles chez *bellicositermes natalensis* et *cubitermes sp.* la théorie de la stigmergie: essai d'interprétation du comportement des termites constructeurs. *Insectes sociaux*, 6:41–80, 1959.

[11] A. Iaminitchi, M. Ripeanu, and I. Foster. Small-world file-sharing communities. In *Proceedings of the 23rd Conference of the IEEE Communications Society (Infocom 2004)*, 2004.

[12] IonStoica, R. Morris, D. Karger, M. F. Kaashoe, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM'01*, August 27-31 2001.

[13] M. Jaczynski and B. Trousse. Www assisted browsing by reusing past navigations of a group of users. In *Proceedings of the European Workshop on Casebase Reasoning, EWCBR'98, LNCS/AI*, Dublin, Irland, September 1998. Springer-Verlag.

[14] H. Lieberman, N. W. V. Dyke, and A. S. Vivacqua. Let's browse : A collaborative web browsing agent. In *Proceedings of the 4th international conference on Intelligent user interfaces (IUT'99)*, pages 65–68, Los Angeles, California, United States, January 05-08 1999.

[15] A. Montresor, H. Meling, and O. Babaoglu. Messor : Load-balancing through a swarm of autonomous agents. Technical Report UBLCS-02-08, Departement of Computer Science, University of Bologna, Bologna, Italy, May 2002.

[16] J. Palau, M. Montaner, and B. López. Collaboration analysis in recommender systems using social networks. In *Eighth International Workshop on Cooperative Information Agents (CIA'04)*, Erfurt (Germany), September 27-29 2004.

[17] S. Ratnasamy. *A Scalable Content-Addressable Network*. PhD thesis, ICSI Center for Internet Research, Berkeley, California., San Diego, California, USA, August 27-31 2002.

[18] C. Schmitz. Self-organization of a small world by topic. In *First International Workshop on Peer-to-Peer Knowledge Management (P2PKM)*, August 2004.

[19] R. Taher. *Recherche d'information collaborative*. PhD thesis, Laboratoire CLIPS-IMAG, Grenoble, 5 Mars 2004.

[20] D. J. Watt and H. S. Steve. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 4 June 1998.

[21] Y. Zhu, H. Wang, and Y. Hu. Integrating semantics-based access mechanisms with p2p file systems. In *Proceedings of the 3rd International Conference on Peer-to-Peer Computing (P2P2003)*, pages 118–125, Linkpings, Sweden, September 2003.