Self-Organization in Service Discovery in presence of Noncooperative Agents

E. del Val , M. Rebollo, V. Botti

Universitat Politècnica de València,
Camí de Vera s/n. 46022, València. Spain
{*edelval,mrebollo,vbotti*}*@dsic.upv.es*

Abstract

Self-organization and cooperation of agents in open societies play an important role in the success of the service discovery process. Self-organization allows agents to deal with dynamic requirements in service demand. Moreover, in distributed environments where service discovery is carried out by agents that only have a partial view of the system, cooperation with neighbors is a key issue in order to locate the required services. However, cooperation is not always present in open agent societies. With this motivation, we present a set of mechanisms that consider self-organization actions and incentives to adapt the structure of the society to the service demand and to promote a cooperative behavior among agents in open societies.

Keywords: Self-organization, cooperation, complex networks, service discovery

1. Introduction

Service discovery systems are deployed in dynamic environments where their components, features, and tasks do not remain constant. These systems are expected to perform well under many circumstances (i.e., when the number of available agents changes, or when the service demand varies with time). For that reason, it is important to take into consideration the inclusion of self-organization mechanisms in order to adapt the social underlying structure to environmental conditions and changes in the requirements [20]. When a global view of the society is not available, the organization process should be performed in a decentralized way without the supervision of any central authority. However, this task becomes even more difficult when there are self-interested agents that do not co-

operate with others. In that case, if there are no mechanisms to deal with these agents and promote cooperation, the performance of the service discovery process could be seriously compromised [9]. The cooperation of entities that participate in a decentralized system and their self-adaptation to environment changes are required to obtain a good performance that provides benefits for all the participants in the context of distributed systems. Some of the scenarios where cooperation and self-adaptation are required are: wireless ad-hoc networks where nodes rely on other nodes to forward their packets in order to reach the destination node; file sharing in P2P systems [24]; streaming applications [16], discussion boards [11], on-line auctions [21], or overlay routing [4].

To illustrate the context where the self-organization and cooperation emergence mechanisms are going to be applied, let us present a service discovery scenario where the service discovery process is described as well as the situations where self-organization mechanisms are applied. Consider a network of services as a form of autonomic cloud computing system. This network contains different groups of semantic web services provided by software agents as part of an overlaying network. These services are provided by agents that, in some situations, should interact with each other to achieve a task that they cannot afford to do individually since they are not specialized in that area or because the task is too complex to be carried out by a single agent. For instance, if an agent i only has local knowledge about the services provided by agents k, n, and j and needs to locate a *payment service* in order to do a bank transaction, it should start a service discovery process to locate the agent v. The service discovery process will require a fewer number of steps if all the agents cooperate that if there are non-cooperative agents. It may happen that agent k and agent n may decide not to cooperate in forwarding messages which will damage the performance of the system. Moreover, if services provided by agent v are being frequently requested by agent i, agent i may consider to adapt its current links in order to reduce the number of steps from agent i to agent v.

In this paper, we present a combination of self-organization and cooperation mechanisms that agents use in order to maintain the performance of the service discovery process when there are changes in the service demand or when selfish agents appear. The self-organization mechanisms focus on how the relations between agents could be rearranged or how the agent population could be adapted according to the service demand to maintain or improve the performance of the service discovery process. The mechanisms that promote cooperation when there are self-interested agents in the society are based on local structural changes and the use of incentives.

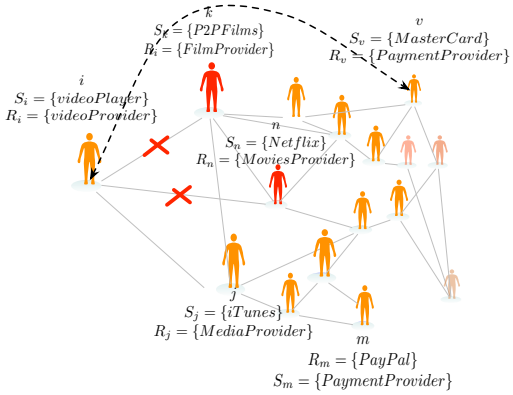


Figure 1: An example of a decentralized service discovery system.

The paper is structured as follows. Section 2 presents works related to the proposal presented in the paper. Section 3 describes the formal model of the system. Section 4 explains how the structure of the network can be modified through local decisions of agents in order to adapt itself to the service demand. Section 5 analyzes the combination of social plasticity and incentive mechanisms in the service discovery scenario to promote cooperation. Section 6 presents a set of experiments where we evaluate the performance of our proposal. Finally, section 7 presents conclusions and final remarks.

2. Related Work

Search approaches commonly used in decentralized systems where all the entities are considered to be equal and there is an arbitrary topology are based on *blind* or *informed* algorithms. *Blind* algorithms do not consider any information about resource locations and use flooding or random strategies that can overload the system with the traffic generated during the the search process [19, 26]. *Informed* approaches try to cope with this problem and consider local information to create and guide the search. The information considered is about their direct neighbors [3, 15] or statistics from previous searches and it is stored in local registries [2].

There are proposals related to decentralized search or service discovery in distributed systems that assume that all the entities that participate in the discovery process are cooperative. However, this not always happen in open societies. We consider an *open society* as a system where different and heterogeneous agents

can join and leave the system. These systems are characterized by heterogeneity of participants, limited trust, different goals, and a high probability of non-conformance to specifications [6]. Approaches based on Game Theory have been widely used to explain mechanisms through which cooperation can emerge and be maintained in different scenarios. Depending on the context, mechanisms such as direct reciprocity [17], indirect reciprocity [18], tags [22], or punishment [12] have been used. Some approaches based on games assume well-mixed populations where everybody interacts with equal frequency with everybody else. However, real populations are not well-mixed. In real populations, some individuals interact more often than others; therefore, to understand the social behavior of the systems it is important to consider the social structure [10].

The approach that we present in this paper for service discovery is based on an informed algorithm that considers local information in order to guide the service discovery process as well as to self-organize the network structure (i.e., the social structure). Initially, the structure is created based on the similarity of the resources provided by the agents. However, the environment conditions do not remain constant. Therefore, in our approach, agents consider self-organization actions in order to maintain or improve the performance of the service discovery process when there are changes in the service demand. Unlike other proposals related to self-organization [25], in our proposal, we consider not only changes in the structure of the agents, but also changes in the population of the system. Moreover, we have considered strategies such as incentives and structural changes to promote cooperation during the service discovery process.

3. Formal Model

Our proposal for agent society is modeled as an undirected network populated by a set of autonomous agents $A = \{i, ..., n\}$ that establish relationships with other agents $L \subseteq A \times A$, where each link $(i, j) \in L$ indicates the existence of a direct relationship between agent i and agent j based on a probability that considers the semantic similarity of their attributes (i.e., the roles and the services of the agents). The role determines the type of services offered by the agent. For a more detailed description of how the structure of the network is created we refer the reader to [8]. An agent is a social entity that interacts with other agents in the society. It controls its own information about:

- the semantic services it offers $S_i = \{s_i, \ldots, s_n\}$;

- the organizational roles it plays $R_i = \{cat_1, \ldots cat_m\}$ where each role is defined by a semantic concept. A role can contain a set of semantic service profile descriptions. Each role has associated a numeric index r_i.

- an internal state st_i that contains local information used by the self-organization and the cooperation mechanisms.

The following information contained in the internal state (st_i) is related to *self-organization* mechanisms:

- N_i is the set of direct neighbors agent i has a direct relationship with. For each neighbor $j \in N_i$, agent i has information about: the roles j plays, the services j offers, the degree of connection of j, and the number of times that a query that arrived to the agent i and was not forwarded through its neighbor j (Q_{ij});

- Acc a set of acquaintances whose existence agent i is aware as a result of the discovery process but it does not have a direct relationship with;

- $\overrightarrow{q_i} = [q_i^{r_1}, q_i^{r_2}, ...,]$ is the local view of the service demand distribution (i.e., the number of queries that the agent receives about services offered by different roles r_1, r_2, ...);

- the *status* of the agent. An agent can be in a *stable* or *transition* status. The status depends on the accuracy of the information about the service demand of the system an agent has. The degree of accuracy is determined by a correlation parameter ρ_i. This parameter establishes the relationship between the local service demand distribution ($\overrightarrow{q_i}$) and the expected service demand distribution. Power-law, Exponential, and Zipf's-law distributions are present in many features of Internet [1, 13, 5]. In our system, the exponential distribution has been considered as the function that models the service demand in the system, where there are always a few services that are the most demanded and the rest of the services have a lower demand rate. If an agent has an accurate view of the service demand (ρ_i close to 1), it is considered to be in a *stable* situation. When a new agent arrives to the system, or when it has information that introduces noise in its local environment, the agent is considered to be in a *transition* situation.

The information in the internal state of an agent (st_i) related to the *cooperation* is:

- dc_i represents the degree of cooperation of agent i and it ranges in the interval [0,1],
- $\mathcal{B}_i$ represents the behavior of agent i. The behavior of the agent can be cooperative or non cooperative and it is established taking into account the agent's payoff and the behavior of its neighbors.
- $\mathcal{F}_i$ is the number of queries that agent i forwarded,
- $\mathcal{SQ}_i$ is the number of queries that went through the agent i and finally arrived to the target agent,
- $\mathcal{RQ}_{ij}$ is the number of queries from agent i that a neighbor agent j refused to forward,
- $\mathcal{P}_i$ is the number of queries solved agent i,
- $\mathcal{C}_i$ is the number of queries created by agent i.

The decision making process about self-organization actions or actions related to the promotion of cooperation is integrated in the service discovery process. The service discovery process follows the steps described below:

1. The process starts when agent i requires the services of other agent in order to achieve one of its goals. Then, agent i creates a query $q = (id, t, TTL, \varepsilon)$, which contains the identifier of the agent that creates the query, semantic profile description of the desired provider agent, the Time To Live of the query, and the similarity threshold that determines when a provider agent is going to be considered enough similar to the target agent. A *target* agent profile description t consists on service s_q and the role r_q that the target agent should play (s_t, r_t).
2. Agent i looks for a neighbor similar to t. Specifically, q is forwarded to the neighbor that has semantic closeness to the *target* agent t and also has a high degree of connection.
3. The selected neighbor j analyzes, based on its payoff and the payoff of its neighbors, if it is worthwhile forwarding the query (i.e., if it is worthwhile to cooperate or not). If j rejects forwarding the query, agent i updates the number of times that neighbor j rejects its request of forwarding ($\mathcal{RQ}_{ij}$). Based on this information, agent i considers breaking its current link with j (see Section 5).

4. If agent j does not cooperate, agent i repeats the steps 2 and 3 until it finds an agent that cooperates. Once a cooperator neighbor is found, agent i forwards the query to it and updates its information about which of its links have been used and which not (Q_{ij}). Agent i also updates the number of total queries it received (Q_i), and the number of queries received about the role r_q ($\overrightarrow{q_i}[r_q]$). In the case that there are no cooperators in the neighborhood, the discovery process fails.
5. When the query reaches a suitable provider agent similar to t, the service discovery process ends and all the participants receive a reward. The agent that started the service discovery process adds the provider agent found to its set of acquaintances (Acc) only if it does not already have an acquaintance that plays the role of the provider agent. Finally, the agent that started the process analyzes its internal state and the set of self-organization actions that it can carry out (see Section 4).

In the following sections, we describe with more detail the actions included in the service discovery process that are related to self-organization and cooperation emergence mechanisms.

4. Self-Organization Mechanisms

In order to make decisions about self-organization actions agents need to have an accurate local view of the service demand in the system. To evaluate the accuracy of their local view agents analyze its internal state (st_i). Initially, an agent is in a *transition* state. An agent in this state does not have reliable and sufficient information to be able to estimate the current service demand distribution in the system according to its local view. In this state, an agent can reorganize its local view of the service demand distribution $\overrightarrow{q_i}$ taking into account the number of queries received about the services associated to each role.

An agent in the *transition* status has three options:

- it changes its status from *transition* to *stable* when ρ_i (i.e., the correlation degree between its local data about service demand in the society, $\overrightarrow{q_i}$, and an estimation of the service demand distribution) is over a threshold δ and it has enough information about the service demand (Q_i) (see Figure 2).

- it remains in the *transition* status if it considers that it has not enough information about the service demand (Q_i) and (ρ_i) is not over a threshold δ.

- it *resets* its local service demand view if it has a big enough deviation of the correlation degree although it has received a high number of queries. This usually happens when there is a change in the service demand. This fact introduces new information in the local view $\overrightarrow{q_i}$ of the agent that does not follow the expected distribution. Therefore, it is important to detect these changes in order to reset the local view of the system with outdated information and only consider the new information that arrives about the new service demand.

The correlation parameter ρ_i indicates the degree of fitness between the local data $\overrightarrow{q_i}$ and the expected exponential distribution $eDistr(x)$, where the x parameter represents a numeric role identifier. In our system, the estimation of the service demand distribution follows an exponential distribution. This type of distribution is present in many features of open systems such as Internet [1, 13, 5]. Specifically, we assume that the expected service demand distribution is $eDistr(x) = a \cdot e^{x \cdot b}$. We estimate the a and b parameters of this distribution using the least squares method and the data from $\overrightarrow{q_i}$. In order to analyze if the agent received a sufficient number of queries we use logistic function (see Equation 1).

$$P(Q_i) = \frac{1}{1 + e^{\frac{-(Q_i - d)}{y}}}, \tag{1}$$

where y is the slope, d is the displacement constant, and Q_i is the number of queries the agent has received. The most influential constant is d. A higher value of d means that the agent is going to consider a higher number of queries in order to make a decision about resetting the information about local view of the service demand $\overrightarrow{q_i}$ or changing to the *stable* state. The function $P(Q_i)$ returns a value in the range [0,1], where 0 indicates that the agent has not received a sufficient number of queries, and 1 indicates that the number of queries is significant.

An agent that is in a stable state resets its local view of service demand and turns back into the *transition* status if it has a big enough deviation of the correlation degree at any moment. This means that the service demand is changing and that the local view of the agent is not accurate with respect to the current systems demand, and it is advisable to reset its local view since the consideration of outdated information introduces noise in the local view of the agent and affects the self-organization process.

Self-organization of the structural links. Once the agent is in an *stable* state, it is able to make decisions about self-organization actions. Agents are able to

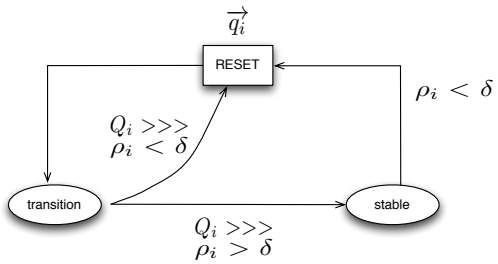


Figure 2: Conditions required to reset the local view of the service demand in its internal state.

reason about whether or not maintain, reinforce or create new structural relations. Agents consider a *decay* metric associated to each link: $decay(Q_{ij}) = 1-(1/(1+e^{-(Q_{ij}-z)/y)}$, where y is the slope, z is the displacement constant, Q_{ij} is the number of queries that arrived to agent i and were not forwarded through neighbor j. This metric ranges in the interval [0,1], where 0 indicates that the link is not being used and 1 indicates the the link is being used. Each time agent i forwards a query, it updates the information about the traffic of its links. If the query is forwarded through agent j, the Q_{ij} is updated to 0. Otherwise, the Q_{ij} is increased by increments of 1. With the information provided by the *decay* function, agent i reasons about the benefit of maintaining its current links.

Population self-organization: leaving, remaining, or cloning. The analysis that evaluates whether it is worthwhile for the agent to remain in the system, clone itself, or leave the system takes the following parameters into account:

- the number of queries received by the agent Q_i;
- the status of the agent;
- SH_i is an indicator that measures the demand of the services provided by an agent i. SH_i ranges in the interval [0,1], where 1 indicates that the services the agent offers are required in the system, and 0 indicates that the services the agent offers are not being demanded in the system. This metric reflects how important an agent is to the system with regard to the current service demand. The structural similarity of an agent with respect the system dynamics is defined by the following function:

$$SH_i = a \cdot e^{r_i \cdot b},$$

where r_i is the numeric index that represents the role of agent i that maximizes the function $a \cdot e^{x \cdot b}$ that represents the estimation of the service demand distribution:

$$r_i = \underset{x \in R_i}{\operatorname{argmax}}\, a \cdot e^{x \cdot b}$$

In the last function, the a and b parameters are obtained through the least squares method and the data from the local view of the service demand $\overrightarrow{q_i}$. An example of how the structural homophily of agent i is calculated is shown in Figure 3. Agent i plays two roles r_1 and r_4. At that moment, using the data in $\overrightarrow{q_i}$, the exponential function that estimates the service demand distribution is $eDistr(x) = 1.73 \cdot e^{x \cdot 1.16}$. Therefore, the structural homophily of agent i is:

$$SH_i = 1.73 \cdot e^{r_1 \cdot 1.16} = 0.54$$

where

$$r_1 = \arg \max_{r_1, r_4 \in R_i} [1.73 \cdot e^{-r_1 \cdot 1.16}, 1.73 \cdot e^{-r_4 \cdot 1.16}] = \arg \max_{r_1, r_4 \in R_i} [0.54, 0.018]$$

This means that the services that agent i offers are being demanded, but are not the most demanded services in the system.

- the number of queries forwarded since the last analysis Δq^i;
- the degree of correlation ρ_i.

First, an agent evaluates whether or not it is worthwhile to remain in the system. The analysis of the *leave* action is based on the following parameters: *number of queries received* Q_i, the *degree of correlation* ρ_i, and SH_i (see Table 1). If the number of queries received is high enough, ρ_i is over a threshold σ, and SH_i has a value near 0; then the agent decides to leave the system. However, this does not always happen. In order to ensure the availability of a certain type of services in the system, the agent does not leave the system if there is no similar neighbor that provides similar services. The number of similar neighbors is calculated by *SimN* function. This function returns the number of agents in the neighborhood that are similar to agent i. This function is based on the semantic similarity between agents described in Appendix A. Finally, if the agent leaves the system, it breaks all the connections with all its immediate neighbors and communicates that it is going to leave. The neighbors will try to find an alternative neighbor based on the semantic similarity.

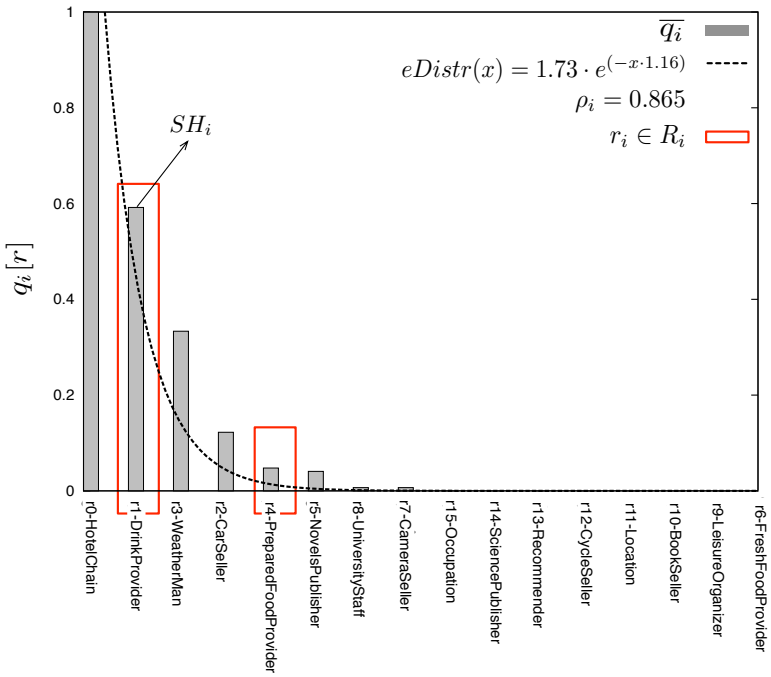


Figure 3: Example of the internal state st_i of the agent i.

If the agent has decided not to leave the system, it analyzes the *clone* action. This analysis is also based on the parameters described above. The main difference is the logistic function for evaluating the significance of the *number of queries received*. This function is similar to function 4. However, in this function, the displacement parameter d takes into account the number of clones that an agent has. If the number of queries received is high enough, ρ_i is over a threshold, and SH_i has a value near 1; then the agent decides to execute the *clone* action. However, this does not always happen. In order to prevent the number of clones increasing exponentially, there are two more conditions that reduce the probability of cloning. The agent does not clone if all its neighbors are similar to it or if the number of queries it forwards has not increased since the last analysis. Taking into account all these parameters, the agent evaluates whether or not creating a clone is worthwhile. The clone generated by the agent will offer the same services and play the same roles, and the number of clones it has will be initialized with the value of its father. The cloned agent establishes links with other agents in the system taking into account the semantic similarity criterion. When an agent

	Leave	Clone
Num. Queries	$\frac{1}{e^{-(Q_i - d')/y}}$	$\frac{1}{(1+\cdot e^{-(Q_i - 2^{clones})/y}}$
Status	Stable	Stable
SH_i	$SH_i < rand(0,1)$	$SH_i > rand(0,1)$
$SimN(N_i)$	$SimN(N_i) > 0$	$SimN(N_i) < \|N_i\|$
Δq^i	$\Delta q^i > 0$	
ρ_i	$\rho_i > \delta$	$\rho_i > \delta$

Table 1: Parameters and conditions that agents use during the making decision process about self-organization actions. The parameters are: the number of queries received by the agent, the status of the agent, the structural homophily SH_i, the similarity of the neighborhood SimN(N_i), the increase in the number of queries received Δq^i, and correlation value ρ_i.

creates a clone it resets its internal state (st_i).

5. Incentives and Social Plasticity

Agents that participate in the service discovery can be cooperative or non-cooperative. Cooperate in the service discovery scenario implies that an agent is going to: forward queries, request services, and attend to request about its services. If an agent has non-cooperative behavior, it means that the agent is going to act selfishly by requesting services and offering its services, but it is not going to forward the queries that it receives from its neighbors. We assume that each action in our model implies a cost and, in order to promote cooperation, the forwarding action has a reward if the query arrives to a suitable provider agent. Otherwise, the agents lose their investment in the forwarding process. Moreover, an agent that locates the required provider agent must pay for the service and the provider gets a benefit for attending to the request.

Agents in a neighborhood share information about their payoffs. An agent establishes its behavior based on its payoff and the payoff of its neighbors. An agent calculates its payoff as follows: $\mathcal{PO}(st_i) = \mathcal{SQ}_i \cdot sq - \mathcal{F}_i \cdot f + \mathcal{P}_i \cdot p - \mathcal{C}_i \cdot c$, where $\mathcal{SQ}_i, \mathcal{F}_i, \mathcal{P}_i, \mathcal{C}_i$ is the information contained in the internal state (st_i) of an agent; sq is the benefit obtained by the agents that participate by forwarding queries in a service discovery process that ends successfully; f is the cost of forwarding queries; p is the benefit obtained by the agents that provide a service; c is the cost of requesting a service. We assume that all the agents have the same costs and benefits for the actions. Agents are rational entities that update their

own behavior to maximize their own payoff.

The strategy followed by the agents in order to change their behavior is based on imitation [23]. Agents take into account the payoff of their direct neighbors to update their behavior. If an agent has a neighbor that obtains a higher payoff, the agent changes its behavior to the behavior of its neighbor.

When the number of cooperative agents is greater than the number of non-cooperative agents, non-cooperative agents are prone to change their behavior to cooperate since the probability that a query ends successfully is high, and, therefore, cooperation receives a reward if the discovery process ends successfully. However, when the number of non-cooperators is greater than the number of cooperators, cooperative behavior does not always emerge. In order to facilitate the emergence of cooperation in this scenario, in our proposed model, each agent also has the capacity to change its relationships as time passes taking into account which neighbors provide profitable relationships and which do not. This feature is called *social plasticity* [10]. Social plasticity is the capacity of individuals to change their relationships as time passes. The social plasticity of the agents in our model is based on a logistic function (see Equation 2) that depends on the number of times a neighbor has refused to forward one of its queries (i.e., it has a non-cooperative behavior). An agent i maintains a counter per each direct neighbor j ($\mathcal{RQ}_{ij}$) that stores the number of times a neighbor rejected forwarding a query [10]. If a neighbor j decides to change its behavior and forwards queries, the agent updates its counter ($\mathcal{RQ}_{ij}$) to 0.

$$D(\mathcal{RQ}_{ij}) = \frac{1}{1 + e^{\frac{-(\mathcal{RQ}_{ij} - d)}{y}}}, \tag{2}$$

In order to find a trade-off between the number of structural changes and the emergence of cooperation, the use of the social plasticity mechanism is affected by the number of non-cooperator neighbors an agent has in its neighborhood. If the number of non-cooperator neighbors is over a certain threshold, the mechanism used to facilitate the emergence of cooperation is the social plasticity mechanism combined with the mechanism based on incentives. Otherwise, the mechanism used is based on incentives only.

With the combination of the social plasticity and incentives, non-cooperative agents lose connectivity, benefits, and influence in the neighborhood. As a consequence, they decide to change their behavior to the most promising behavior in the neighborhood, which is to cooperate.

6. Experiments

We analyzed the effects of using of self-organization and cooperation mechanisms in the discovery process. The tests were performed on a set of 20 undirected networks with an average degree of connection of 4. The degree of connection distribution follows an exponential distribution. The creation process of the network is described with detail in [8]. The networks were populated by 1,000 agents. The agents played one role and offered one semantic web service associated to this role. Initially, the agents were uniformly distributed over 16 roles, which were defined in an organizational ontology. The set of semantic service descriptions used for the experiments was taken from the OWL-S TC4 test collection [1].

All the agents in the system had the same probability of generating service queries. A query consisted of the identifier of the agent that creates the query, semantic profile description of the desired provider agent, the Time To Live of the query (TTL), and the similarity threshold ε that determines when a provider agent is going to be considered enough similar to the target agent. A query was successfully solved when an agent that offered a similar service (i.e., the degree of semantic match between the semantic service descriptions was over a threshold $\varepsilon = 0.75$) was found in a number of steps lower than the TTL of the query ($TTL = 100$). Query distribution in the system was modeled as an exponential distribution. In the experiments, we made a snapshot of the metrics every 10,000 queries in order to see the evolution of the system.

Specifically, the tests focused on a set of metrics that are meaningful for the analysis of the performance of the system and for the effects on the service discovery process when agents incorporate self-organization and cooperation mechanisms [14]. These metrics are: (i) evolution of cooperation in the system; (ii) number of broken relationships as consequence of social plasticity; (iii) the evolution of agents' self-organization through changes in the links and in the population; (iv) average number of steps required to locate an appropriate agent that solves a query; and (v) % of queries that are solved before the TTL.

The values of the parameters used in the experiments are shown in Table 2. The values of these parameters have been analyzed in a previous work [7]. The results were evaluated considering two different configurations. In one configuration, the initial number of cooperators in the network was 600. In the other configuration, the initial number of cooperators was 400. We compare the results that we obtained using the proposed mechanisms with the results obtained in static

[1] http://www.semwebcentral.org/projects/owls-tc/

Action	Equation	Param. value
leave	$1/e^{-(Q_i - d')/y}$	$d' = 200, y = 4$
clone	$1/(1 + \cdot e^{-(Q_i - 2^{clones})/y})$	$y = 4$
states	$\rho_i < \delta$	$\delta = 0.7$
links	$decay(Q_{ij}) = 1 - 1/(1 + \cdot e^{-(Q_{ij} - z)/y})$	$z = 300, y = 4$
costs/rewards	$\mathcal{PO}(st_i)$	$q = 0.15$, $p = 0.5$, $r = 0.5$, $sq = 0.30$
social plasticity	$D(\mathcal{RQ}_{ij}) = 1/(1 + e^{-(\mathcal{RQ}_{ij} - d)/y})$	$d = 7, y = 1$

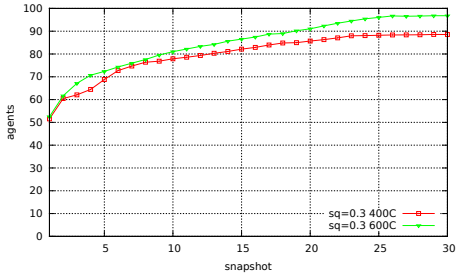
Table 2: Parameters' values use during the making decision process about self-organization actions and cooperation emergence.

networks without these mechanisms. We considered three scenarios. In the first scenario, the initial distribution of roles and services is uniform. In the second scenario, the initial distribution of roles and services follows a normal distribution. Finally, in the third scenario, we modify the degree of connection of the agents.

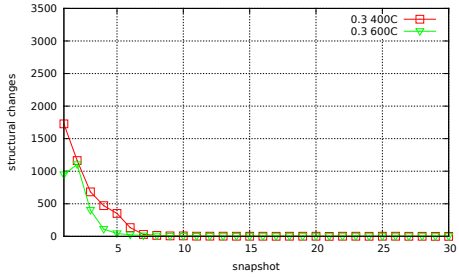
6.1. Uniform distribution of roles

In this scenario, the initial distribution of roles and services over agents follows a uniform distribution and we analyze the effects of having an initial population distribution that does not follow the same distribution as the service demand and where there are non-cooperator agents.

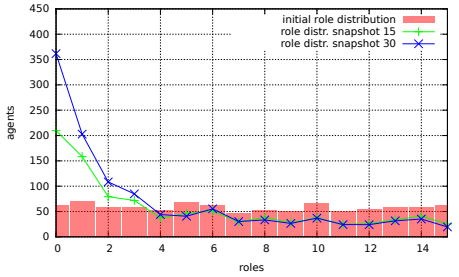
Figure 4a shows the evolution of cooperators when the initial number of cooperators was 400 (or 600) and the type of services and roles are distributed uniformly over the agents. In the first snapshots, the number of cooperators increases at a higher rate than in the following snapshots. The main reason is that when agents start to interact in the service discovery process, they realize that there are neighbors that reject forwarding the queries. As consequence, agents use social plasticity to isolate these selfish agents that do not cooperate (see Figure 4b). Once the number of non-cooperative neighbors has been reduced, the use of social plasticity it is not required and the use of incentives is enough to change the behavior of non-cooperative agents. The use of incentives promotes cooperation but requires a higher number of searches than social plasticity to promote it. The main advantage of incentives is that this mechanism does not break links that could disconnect or reduce the connectivity of the network. When the initial distribution of services follows a uniform distribution, there is not a big difference between the configuration where there are 600 or 400 cooperative agents. The main difference



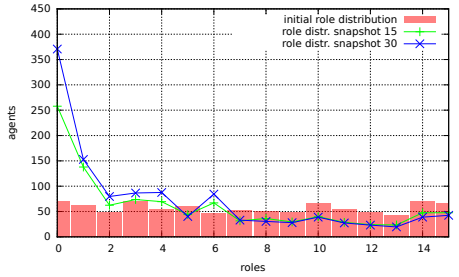
(a) % of agents that collaborate.



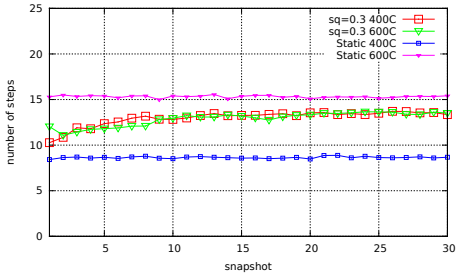
(b) Social Plasticity.



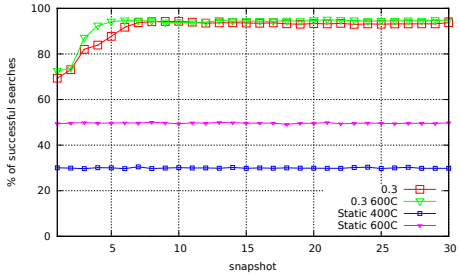
(c) Adaptation (400 inital cooperators).



(d) Adaptation (600 inital cooperators).



(e) Mean Path.



(f) Success.

Figure 4: Effects of the combination of self-organization and cooperation mechanisms in the service discovery process when the initial distribution of agents follows a uniform distribution ($\mu = 8, \sigma = 3$) and the average degree of connection of the agents is 4. We considered two configurations. In one configuration the number of initial cooperators in the network was 600. In the other configuration the initial number of cooperators was 400.

is that the social plasticity required when there are less cooperator agents is higher. Therefore, the number of agents that can be isolated from the giant component is higher and for that reason, in the last iterations there are always agents that remain with the non-cooperative behavior.

Figure 4c and 4d show the structural adaptation of the agents as well as the adaptation of the population. We considered the initial role distribution, the distribution of roles in the snapshot 15, and the final distribution of roles in snapshot 30. In general, the self-organization mechanisms are able to adapt the structure to the service demand. Figure 4c shows the adaptation when the initial cooperators were 400 agents and Figure 5d when the initial cooperators were 600. It can be observed that the adaptation is influenced by the degree of cooperation in the system. The self-organization requires a higher number of search processes in the configuration where there were 400 cooperators than in the configuration of 600 cooperators.

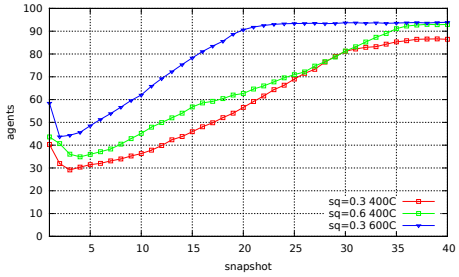
Figure 4e shows the average number of steps in successful searches. In the case of 600 initial cooperators, the introduction of self-organization mechanisms decreased the number of steps required to reach a suitable provider agent if we compare them with the steps required when the network was static and incentives were not considered. In the other configuration, where the initial number of cooperators was 400, the average number of steps increased if we compare it with a static network. It makes sense since in static networks with 400 cooperators the only successful queries were those that were solved in the neighborhood of the agent that generated the query.

Figure 4f shows the effects of using self-organization and cooperation mechanisms in the success of the service discovery process. In general, the percentage of queries that ended successfully was improved with the inclusion of the mechanisms. This improvement was achieved in the first snapshots where the self-organization and the promotion of cooperation played an important role.

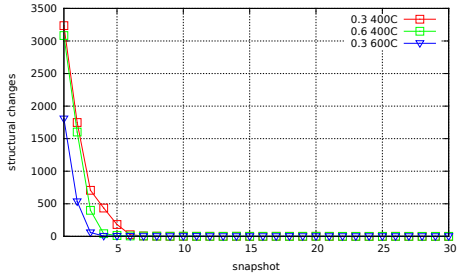
6.2. Normal distribution of roles

In this scenario, we considered a different initial distribution of services and roles over agents in order to see the effects in the self-organization and cooperation mechanism. Moreover, we also considered an increase on the reward received by the agents ($sq = 0.6$) when the number of initial cooperators was 400 in order to see the influence of an increase in the incentives.

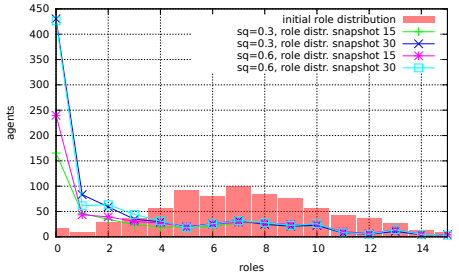
Figure 5a shows the evolution of cooperation when the initial number of cooperators was 400 (or 600) and the roles of the agents were distributed following



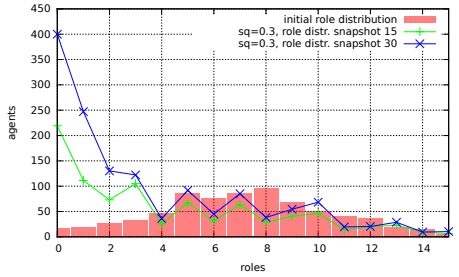
(a) % of agents that collaborate.



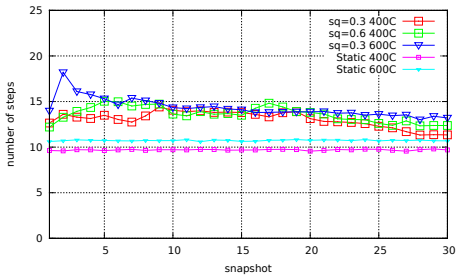
(b) Social Plasticity.



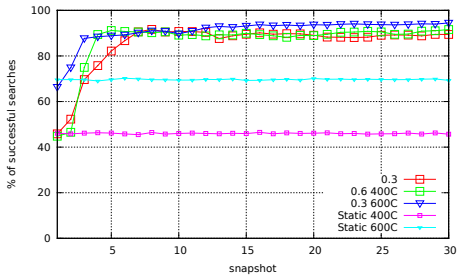
(c) Adaptation (400 inital cooperators).



(d) Adaptation (600 inital cooperators).



(e) Mean Path.



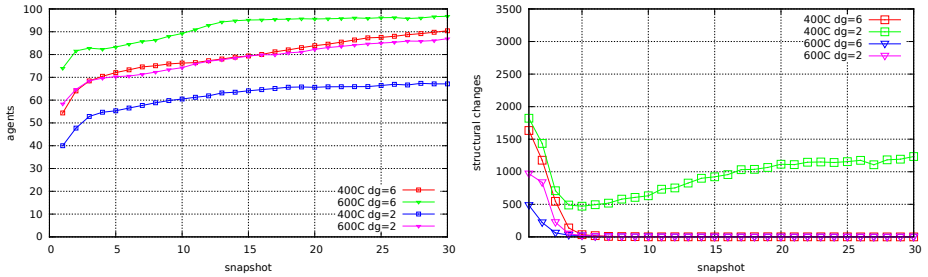
(f) Success.

Figure 5: Effects of the combination of self-organization and cooperation mechanisms in the service discovery process when the initial distribution of agents follows a normal distribution ($\mu = 8, \sigma = 3$) and the average degree of connection of the agents is 4. We considered two configurations. In one configuration the number of initial cooperators in the network was 600. In the other configuration the initial number of cooperators was 400.

a normal distribution (see Figures 5c and 5d). Initially, the percentage of cooperative agents decreases. This effect is caused by the presence of non-cooperative agents as well as the network is not adapted to the service demand (i.e., there is a low number of agents that provide the most demanded services). These conditions make that the probability of reach a suitable provider decreases, and therefore, the probability of receiving a reward. However, in the first interactions between agents, they detect that there are selfish agents that do not forward their queries and they start to isolate selfish agents through the use of social plasticity (see Figure 5b). Social plasticity is a mechanism that it is useful in scenarios where there are a high number of non-cooperative agents. As the number of non-cooperators is reduced, the use of social plasticity decreases. The social plasticity plays a key role in the initial configuration where there are only 400 agents that cooperate. The number of structural changes caused by the social plasticity is higher than the changes required in the configuration where there were 600 cooperators. It can be also observed in the configuration with 400 cooperators that the increase in the reward that the agents received reduces the number of structural changes as well as improves the cooperation degree.

Figures 5c and 5d show how the self-organization mechanisms adapt the population of the network in order to deal with a change in the service demand. The number of agents that play the roles r_0, r_1, and r_2 is higher than in the previous scenario since the initial number of agents that provided was lower than in the uniform distribution. Therefore, the agents that offered the most demanded services received a high number of queries and they considered the clone action earlier facilitating the adaptation to the service demand. As in the previous scenario, the cooperation influences the degree of adaptation. If the number of cooperator agents is high, the degree of adaptation is higher.

The effects of the integration of self-organization and cooperation emergence mechanisms in the service discovery process are shown in Figures 5e and 5f. Regarding the mean path of the service discovery process, initially, the number of steps required to reach a suitable provider increases. This fact is due to the presence of a high number of non-cooperator agents and the initial normal distribution of agents (i.e., there is a low number of provider agents that offer the most demand services). As the system is self-organized according to the service demand, the average path decreases. The main difference between the configuration of 400 initial cooperators and the configuration of 600 cooperators is that in the last one, in the first snapshots, the average path length consists on more steps due to the cooperation is higher and it is possible to find providers that are farther away. Regarding the success rate, the self-organization and the promotion of cooperation improves



(a) % of agents that collaborate.

(b) Social Plasticity.

(c) Mean Path.

(d) Success.

Figure 6: Effects of the combination of self-organization and cooperation mechanisms in the service discovery process when the initial distribution of agents follows a uniform distribution and the average degree of connection of the agents is 6. We considered two scenarios. In one configuration the number of initial cooperators in the network was 600. In the other configuration, the initial number of cooperators was 400.

the number of successful queries most of all in the first snapshots.

6.3. Influence of the average degree of connection

In this scenario, we change the average degree of connection of the agents in the system in order to see its influence in the self-organization and cooperation emergence mechanisms and, therefore, in the service discovery process. In general, the results show that an increase in degree of connection has an important influence in the emergence of cooperation. An increase in the degree of connection means that the number of paths between two agents in the network increases. As there are more alternative paths between two agents, the effect of non-cooperative agents is not so important. Agents are able to find alternative paths to reach the target agent, therefore, there is a higher probability to obtain a reward for partic-

ipating in the discovery process. Agents with higher benefit are those who cooperate and the non-cooperative agents change their behavior to imitate them. The emergence of cooperation in the first snapshots is greater in networks with high degree of connection than in networks with lower degree of connection (see Figure 6a). As the degree of connection increases, the use of social plasticity decreases since there number of neighbors increases and the proportion of non-cooperative agents in the neighborhood is reduced (see Figure 6b). We do not show the results about the evolution of the adaptation since there is not a significant difference in the adaptation of the population of the network to the service demand due to the number of agents is the same. As the increase in the average degree of connection of the agents improves cooperation, the success rate of the discovery process is also improved (see Figure 6d).

7. Conclusions

Our proposal addresses the problem of self-organization and cooperation of agents in order to deal with the service discovery when service demand changes or selfish agents appear in open societies. Agents include *self-organization* mechanisms in order to adapt the underlying structure of the agent society to changes in the service demand. Agents replace their relationships with neighbors that are not being used with new structural relations with acquaintances. Agents are also able to estimate whether or not they are playing an important role in the society through the calculation of the similarity between the services they offer and the services that are being demanded in the system. With this information, agents decide to remain, leave, or clone themselves in order to adapt the population to the service demand. We also include the use of incentives and social plasticity in order to promote and maintain *cooperation* in the society. Incentives influence the behavior of other agents and promote cooperation. Moreover, social plasticity allows agents to change their structural relations based on the degree of cooperation of their neighbors. We evaluated the integration of the proposed mechanisms in different scenarios through a set of experiments taking into account the effects on the evolution of cooperation, the degree of adaptation of the system structure to the service demand, the average path length of a service discovery process, and the percentage of successful searches. The results show that the proposed mechanisms improve the service discovery performance increasing the success, reducing the path length, and increasing the number of cooperators in the agent society.

Acknowledgements

This work is supported by TIN2011-27652-C03-01 and TIN2012-36586-C03-01 projects and PROMETEOII/2013/019, the Program Valorización y Recursos Conjuntos de I+D+i VLC/CAMPUS and the Ministerio de Educación, Cultura y Deporte as part of the Campus of International Excellence Program SP2014800.

Appendix A. Calculation of Semantic Similarity of Agents

In the presented system, the similarity of two agents i and j is based on the degree of match between two sets of services $M_S(\mathcal{S}_i,\mathcal{S}_j)$ and the degree of match of the organizational roles $M_R(R_i, R_j)$. The parameters $\mathcal{S}_i$ and $\mathcal{S}_j$ are the sets of services provided by the agents i and j, respectively. The parameters R_i and R_j are the sets of roles played by the agents i and j.

$$[Sim(i,j) = (1-\varphi) * M_S(\mathcal{S}_i, \mathcal{S}_j) + \varphi * M_R(R_i, R_j)]$$

The φ parameter regulates the importance of the influence of services or roles in the total similarity of the agent with another agent.

For the calculation of $M_S(\mathcal{S}_i,\mathcal{S}_i)$, we consider each set of services $\mathcal{S}_i$ (or $\mathcal{S}_j$) to be composed of a set of semantic concepts that can be classified as: Inputs (I_i), Outputs (O_i), Preconditions (P_i), and Effects (Eff_i). The level of matching between two sets of semantic concepts, C_i and C_j, is calculated through a *bipartite matching graph* ($W_{G'}$).

Specifically, to calculate the similarity between two agents, four bipartite graphs are defined, (one for each of the components of services present in the sets $\mathcal{S}_i$ and $\mathcal{S}_j$): Inputs (I_i, I_j), Outputs (O_i, O_j), Preconditions (P_i, P_j), and Effects (Eff_i, Eff_j). The linear combination of the $W_{G'}$ of each set of concepts gives the value of the homophily between agents (see Equation Appendix A, where the parameters α and β assign different weights to the components of the equation).

$$M_S(\mathcal{S}_i, \mathcal{S}_j) = \alpha\big[\beta * W_{G'_I} + (1-\beta)W_{G'_O}\big] + (1-\alpha)\big[\beta * W_{G'_P} + (1-\beta)W_{G'_{Eff}}\big]$$

The degree of match $M_R(R_i, R_j)$ between the set of roles R_i and R_j played by the agents i and j is defined as the maximum degree of match between the semantic concepts cat_i and cat_j that describe the roles $r_i \in R_i$ and $r_j \in R_j$ for all possible pairs (r_i, r_j).

$$M_R(R_i, R_j) = \max_{r_i \in R_i, r_j \in R_j} rmatch(cat_i, cat_j)$$

The value obtained in the calculation of $M_S(\mathcal{S}_i,\mathcal{S}_j)$ and $M_R(R_i, R_j)$ range in the interval [0,1], where 1 indicates that the services/roles are the same. For a more detailed description of the calculation of semantic similarity of agents we refer the reader to [8].

[1] Adamic. Zipf's law and the internet. *Glottometrics*, 3:143–150, 2002.

[2] U. Basters and M. Klusch. Rs2d: Fast adaptive search for semantic web services in unstructured p2p networks. In *Proc. of the ISWC*, volume 4273, pages 87–100. Springer, 2006.

[3] D. Bianchini, V. D. Antonellis, and M. Melchiori. Service-based semantic search in p2p systems. volume 0, pages 7–16, Los Alamitos, CA, USA, 2009. IEEE Computer Society.

[4] A. Blanc, Y.-K. Liu, and A. Vahdat. Designing incentives for peer-to-peer routing. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 1, pages 374 – 385, march 2005.

[5] M. Costa and M. J. Silva. Characterizing search behavior in web archives. In *TWAW*, pages 33–40, 2011.

[6] M. Dastani, V. Dignum, and F. Dignum. Role-assignment in open agent societies. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '03, pages 489–496. ACM, 2003.

[7] E. Del Val. *Semantic Service Management for Service-Oriented MAS*. PhD thesis, Departament de Sistemes Informàtics i Computació, Universitat Politècnica de València, 2013.

[8] E. Del Val, M. Rebollo, and V. Botti. Enhancing Decentralized Service Discovery in Open Service-Oriented Multi-Agent Systems. *JAAMAS*, pages 1–30, 2013.

[9] J. E. Doran, S. Franklin, N. R. Jennings, and T. J. Norman. On cooperation in multi-agent systems. *The Knowledge Engineering Review*, 12:309–314, 1997.

[10] V. M. Eguiluz et. al. Cooperation and emergence of role differentiation in the dynamics of social networks. *American Journal of Sociology*, 110:977, 2005.

[11] B. Gu and S. Jarvenpaa. Are contributions to p2p technical forums private or public goods? - an empirical investigation. In *Proceedings of the 1st Workshop on Economics of Peer-to-Peer Systems*, 2003.

[12] C. Hauert, A. Traulsen, H. Brandt, M. A. Nowak, and K. Sigmund. Via Freedom to Coercion: The Emergence of Costly Punishment. *Science*, 316(5833):1905–1907, 2007.

[13] B. A. Huberman and L. A. Adamic. The nature of markets in the www. Technical report, 1999.

[14] E. Kaddoum, C. Raibulet, J.-P. Georgé, G. Picard, and M.-P. Gleizes. Criteria for the evaluation of self-* systems. *Proc. of the SEAMS*, pages 29–38, 2010.

[15] D. Kontominas, P. Raftopoulou, C. Tryfonopoulos, and E. Petrakis. Ds4: A distributed social and semantic search system. In *Advances in Information Retrieval*, volume 7814 of *Lecture Notes in Computer Science*, pages 832–836. 2013.

[16] W. Lin, H. Zhao, and K. Liu. Incentive cooperation strategies for peer-to-peer live multimedia streaming social networks. *Multimedia, IEEE Transactions on*, 11(3):396 –3412, april 2009.

[17] M. A. Nowak. Five Rules for the Evolution of Cooperation. *Science*, 314(5805):1560–1563, 2006.

[18] M. A. Nowak and K. Sigmund. Evolution of indirect reciprocity by image scoring. *Nature*, 393(6685):573–577, 1998.

[19] A. Ouksel, Y. Babad, and T. Tesch. Matchmaking software agents in b2b markets. In *Proc. of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04)*, 2004.

[20] M. P. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann. Service-oriented computing: State of the art and research challenges. *Computer*, 40:38–45, 2007.

[21] J. Shneidman and D. C. Parkes. Rationality and self-interest in peer to peer networks. In *Proceedings of the 2nd Int. Workshop on Peer-to-Peer Systems (IPTPS'03)*, 2003.

[22] K. Sigmund. Sympathy and similarity: The evolutionary dynamics of cooperation. *Proceedings of the National Academy of Sciences*, 106(21):8405–8406, 2009.

[23] D. Strang and M. W. Macy. In search of excellence: Fads, success stories, and adaptive emulation. *American Journal of Sociology*, 107:107, 2001.

[24] Q. Sun and H. Garcia-Molina. Slic: A selfish link-based incentive mechanism for unstructured peer-to-peer networks. In *Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'04)*, pages 506–515, Washington, DC, USA, 2004. IEEE Computer Society.

[25] L. Wang. Sofa: An expert-driven, self-organization peer-to-peer semantic communities for network resource management. *Expert Syst. Appl.*, 38(1):94–105, Jan. 2011.

[26] M. Zhong. Popularity-biased random walks for peer-to-peer search under the square-root principle. In *Proc. of the 5th International Workshop on Peer-to-Peer Systems*, 2006.