Promoting Cooperation in Service-Oriented MAS through Social Plasticity and Incentives

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

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

Abstract

In distributed environments where entities only have a partial view of the system, cooperation plays a key issue. In the case of decentralized service discovery in open Service-Oriented Multi-Agent Systems, agents only know about the services they provide and their direct neighbors. Therefore, they need the cooperation of their neighbors in order to locate the required services. However, cooperation is not always present in open and distributed systems. Non-cooperative agents pursuing their own goals could refuse to forward queries from other agents to avoid the cost of this action; therefore, the efficiency of the decentralized service discovery could be seriously damaged. In this paper, we propose the combination of local structural changes and incentives in order to promote cooperation in the service discovery process. The results show that, even in scenarios where the predominant behavior is not collaborative the cooperation emerges.

Keywords: cooperation, service discovery, incentives, complex networks

1. Introduction

There are distributed systems where the cooperation of all the entities that participate in them is required to obtain a good performance that provides benefits for all the participants. Some of the scenarios where cooperation is 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 [48]; streaming applications [29], discussion boards [19], on-line auctions [43], or overlay routing [8]. If participants that decide not to contribute in order to maximize their own benefits and exploit the contributions of the others appear in these scenarios, they will obtain a high rate of benefits in the short term. However, these benefits decrease as the number of selfish participants increases, thereby damaging the performance of the whole system. There are models of genetic and cultural evolution that confirm that the opportunity to take advantage of others undermines and often eliminates cooperation [21]. These cooperation problems are also known as social dilemmas (i.e., the tragedy of the commons, the free-rider problem, the social trap). The promotion and stabilization of cooperation in scenarios of this type has been considered to be an area of interest [13].

One of the scenarios where cooperation plays an important role is service discovery in open Service-Oriented Multi-Agent systems (SOMAS) [11]. These systems are populated by agents that offer their functionality through services [34]. Agents are social entities that are aware of other agents. However, sometimes this awareness is not enough to find potential collaborators in order to achieve the goals of agents. Therefore, open SOMAS should provide mechanisms to facilitate the discovery of services provided by other agents. Nevertheless, this is not an easy task due to the intrinsic characteristics of these systems. As a consequence, agents need the cooperation of their neighbors in order to forward queries to locate the required resources or services. Moreover, this becomes even more difficult when there are self-interested agents that do not cooperate with other agents in order to avoid the cost of forwarding queries. In that case, if there are no mechanisms to deal with these agents and promote cooperation, the performance of the whole system could be seriously compromised.

This paper proposes a combination of decentralized mechanisms to facilitate the emergence of cooperation in a service discovery scenario. In this scenario, agents are located in a network and their interactions are influenced by the network structure. We propose the integration of local structural changes and the use of incentives to promote cooperation when self-interested agents appear. The proposed combination of mechanisms has been tested and compared with other cooperation mechanisms. The results show that even in adverse situations where there is a large number of non-cooperative (non-cooperator) agents our proposal obtains better results than other mechanisms proposed in the literature and the performance of the system is not seriously affected.

The paper is structured as follows. Section 2 presents a review of related work about decentralized service discovery and mechanisms that facilitate the emergence of cooperation. In section 3, we describe the abstract model of the system where the mechanisms are integrated. Section 4 describes the scenario where the proposed mechanisms are used. Section 5 explains how the structure of the network can be modified through local decisions of agents about their neighbors to facilitate the emergence of cooperation in the service discovery scenario. Section 6 analyzes the incentive mechanism in the service discovery scenario. Section 7 presents the combination of structural changes and incentives. Section 8 presents a set of experiments where we evaluate the performance of our proposal and compare it with other other well-known mechanisms. Finally, section 9 presents conclusions and final remarks.

2. Related Work

Nowadays, there is a trend towards the design of open systems that are populated by a large number of entities that interact with each other in order to share their resources or achieve a complex goal. Several approaches have been proposed to facilitate the location task in these systems. Moreover, the success of resource location in distributed system relies on the cooperation behavior of the members involved in the process. For this reason, mechanisms that promote cooperation are gaining importance. In this section we review some of the works in decentralized resource search and in cooperation emergence in distributed environments.

2.1. Service Discovery in Distributed Environments

Search approaches commonly used in decentralized systems for locating services or resources 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 search process [38, 31, 32, 51].

In order to prevent the generation of traffic, *informed* algorithms that consider local information have been proposed [10, 3]. The information is about their direct neighbors or statistics of previous searches. These algorithms require a period of time to collect information that improves the search. The drawback is that if links between peers change frequently, statistical information stored in local indexes could become useless. Moreover, some of the heuristics that are used to guide the search process could overload some peers and leave other potential peers without traffic. Other approaches use biologically inspired techniques to locate and organize resources [16, 33]. For instance, *ant algorithms* are suitable for unstructured networks because they do not rely on global knowledge. Shen et al. [42] present a P2P system that facilitates service composition where the search of peer candidates for the composition is carried out following two approaches: an ant colony optimization and a genetic algorithm. Moreover, this approach also takes into account Quality of Service parameters to evaluate the suitability of services. Similarity, the work presented by Blake et al. [7] also focuses on the use of Quality of Service parameters. However, this work pays attention to service-level agreements (SLA) in service compositions in order to guarantee the Quality of Service to the consumers. The drawback of this proposal is that it is oriented to centralized repositories.

There are other informed approaches where the underlying structure of the system is *loosely structured* using certain criteria. This facilitates the search process [50, 6]. Initially, agents are connected randomly and they use a reorganization algorithm to group agents with similar services together. In order to avoid isolated clusters of agents, these algorithms establish a percentage of similar and dissimilar agents that should be in the neighborhood of the agent. The main disadvantage of these approaches is the high cost of communication required to organize the entities into communities and the consideration of a fixed number of neighbors that should be similar and dissimilar which reduces the flexibility of the system. *Semantics* has been included in several systems as one of the criterion to organize the network structure, to provide new ways of resource location, and to improve the accuracy of the results [4, 6, 3]. The approach presented by Loia et al. [30] proposes the use of a collaborative proximity-based fuzzy clustering algorithm to organize semantic web data that could be used to self-organize a semantic overlay network. Semantic information plays an important role in distributed systems since it provides interoperability and reusability, therefore, ontologies have been also included in agent-oriented methodologies. As an example, the work presented by Beydoun et al. [5] presents an application example of an ontology-based methodology in a P2P environment where agents share information about their resources.

2.2. Cooperation in Distributed Environments

In general, many of the approaches that deal with decentralized search of resources assume that entities that are part of the system have a cooperative behavior. However, in real scenarios this fact cannot be assumed. In this section, we present some of the areas that traditionally have dealt with the cooperation emergence.

Approaches based on Game Theory have been widely used to explain mechanisms through which cooperation can emerge and be maintained in different scenarios. For instance, in scenarios where individuals interact repeatedly, selfish or altruistic actions would be returned in future. In these scenarios, the mechanism to facilitate the emergence of cooperation is *direct reciprocity*. In every round, an individual has two alternatives: to cooperate or not cooperate. If the individual cooperates, the other individual may cooperate later. Hence, it might compensate to collaborate. In this scenario, the best strategy when the majority are defectors is ”tit-for-tat”. Otherwise, the strategy ”win-stay, lose-shift” is better for maintaining cooperation [35]. When agents do not always interact with the same individuals, there are other mechanisms such as *indirect reciprocity* or *tags*. *Indirect reciprocity* is used in environments where agents interact with other agents who have information about their previous interactions with other agents. Trust and reputation are techniques that are used for indirect reciprocity [36]. Mechanisms based on *tags* facilitate the emergence of cooperation [20]. Tags are established taking into account cultural artifacts or traits [46]. *Punishment* has also been considered to promote cooperation and to overcome the ”tragedy of the commons” [21]. Punishment is present in human societies where sanctioning institutions apply a punishment to those that do not obey the law. In systems where such centralized institutions do not exist, individuals are willing to punish defectors even though this implies a cost for them [22]. In general, punishment has been proven to be an efficient way to maintain cooperation [44, 45].

Many approaches that are 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. The social structure is represented by a network where links are established by the individuals following certain preferences. There are several works that analyze the influence of the network structure in the emergence of cooperation. These works study how structural parameters such as clustering or degree distribution affect the emergence and maintenance of cooperation [40, 37, 41, 23].

Although there are many works that take into account the structure of the networks, there are some works that not only consider the structure of the network, but also consider how local changes in the network structure can influence the collective social behavior. Eguíluz et al. [14] present a model that uses the Prisioner's Dilemma game [2] and *social plasticity* in random undirected networks of agents. Agents update their behavior in discrete time steps using an imitation strategy that considers the payoff of neighbors. The social plasticity (i.e., changes in structural links) is considered when an agent imitates a defector in order to facilitate the replacement of an unprofitable relationship with a new one that is randomly chosen. This process creates a hierarchical topology that plays an important role sustaining cooperation. Griffiths et al. [18] propose a mechanism that considers context awareness and tags of agents to promote cooperation. Moreover, agents can remove part of their connections with agents that are not cooperative and add connections with others that can improve cooperation. There are other approaches that also make use of rewiring techniques and partial observation to facilitate the emergence of cooperation [49]. Rewiring allows agents to decide to replace a link after a number of unsuccessful interactions. Partial observation allows agents to observe a subset of other agents that are located outside of their circle of interaction, and afterwards, the agent imitates the majority action taken by the observed agents.

In this paper, our proposal facilitates cooperation in the service discovery process among agents that are located in a network structure that is based on preferences. The combination of mechanisms that we propose is based on social plasticity and incentives. The main differences with existing approaches are the following: (i) our mechanism is asynchronous, i.e., agents update their behavior when they participate in the service discovery process; (ii) the payoff calculation is based on local information obtained from the activity of agents and the results in the discovery process; (iii) in the rewiring process, agents only break links with those neighbors that have non-cooperative behavior, and instead of replacing them randomly, the agents look for another agent based on their preferences; (iv) agents are able to detect when it is more appropriate to use incentives or social plasticity taking into account local information about the degree of cooperation of their neighborhood.

3. Abstract Model and Notation

The proposed combination of mechanisms to facilitate the emergence of cooperation is applied in a network of autonomous agents that offer their functionality through a set of semantic services. These agents have a reduced view of the global community: just a limited number of direct neighbors are known and the rest of the network remains invisible to them. These agents have a set of neighbors that are established based on a social feature called *homophily* [27, 12]. The idea behind the homophily concept is that individuals tend to interact and establish links with similar individuals through a set of social dimensions. In the context of SOMAS, two agents are considered similar if they play similar roles and offer similar services.

The structural relations between agents influence their interactions, and, therefore, influence the behavior of the agents. In the discovery process, if an agent needs to locate another agent that provides a service in order to achieve one of its goals, it should rely on the cooperation of its direct neighbors. Based on the local information about the success or failure of their previous interactions with their direct neighbors, agents update their behavior (i.e., cooperate or not cooperate in the discovery process) and decide when it is appropriate to change their current structural relations. The main components that are part of the system and that participate in the discovery process are described in this section.

DEFINITION **1.** *(System). The system is defined as an Open Service-Oriented Multi-Agent System $SOMAS = (A, L)$, where $A = \{a_i, ..., a_n\}$ is a finite set of autonomous agents that are part of the system, and $L \subseteq A \times A$ is the set of links, where each link $(a_i, a_j) \in L$ indicates the existence of a direct relationship between agent a_i and agent a_j based on their degree of homophily.*

It is assumed that the knowledge relationship between agents is symmetric, so the network is an undirected graph.

In our model, agents are social heterogeneous entities that play an organizational role that determines the type of services an agent offers. Moreover, agents have an internal state where they have information about their degree of cooperation and their activity related to the discovery process.

DEFINITION **2.** *(Agent). In this context, an agent $a_i \in A$ is characterized by a tuple of four elements (OR_i, S_i, N_i, st_i) where:*

- *$OR_i = \{r_1, \dots r_m\}$ is the set of organizational roles played by the agent;*
- *$S_i = \{s_1, \dots, s_l\}$ is the set of semantic service descriptions of the services provided by the agent. Each service should be associated to at least one of the roles played by the agent, $s_i \in \bigcup_{\forall r_i \in OR_i} S_{r_i}, \forall s_i \in S_i$;*
- *N_i is the set of neighbors of the agent, $N_i \subseteq A - \{a_i\} : \forall a_j \in N_i, \exists (a_i, a_j) \in L$, and $|N_i| > 0$. It is assumed that $|N_i| \ll |A|$;*
- *st_i is the internal state of the agent. It is defined by a set of $(dc_i, \mathcal{C}_i, \mathcal{Q}_i, \mathcal{SQ}_i, \mathcal{RQ}_{ij}, \mathcal{P}_i, \mathcal{R}_i)$:*
  - *dc_i represents the degree of cooperation of agent a_i. dc_i ranges in the interval [0,1],*
  - *$\mathcal{C}_i$ represents the behavior of agent a_i. $\mathcal{C}_i$ can take two values: cooperative or not cooperative,*

- *$\mathcal{Q}_i$ is the number of queries that agent a_i forwarded,*
- *$\mathcal{SQ}_i$ is the number of queries that the agent a_i forwarded in successful discovery processes,*
- *$\mathcal{RQ}_{ij}$ is the number of queries from agent a_i that agent a_j refused to forward,*
- *$\mathcal{P}_i$ is the number of service requests attended to by agent a_i,*
- *$\mathcal{R}_i$ is the number of service requests sent by agent a_i,*

In the context of multi-agent systems, the concept of role represents the functional position of an agent. An agent must play at least one role, but it can play several roles. Moreover, a role can be played by several agents. The role determines the type of services offered by the agent. The agent acquires a role defined inside an organization of the system if it satisfies a set of requirements [15]. In our context of service discovery, a role is semantically defined by an ontological concept defined in an organizational ontology and by a set of semantic service descriptions associated to the role that describes the functionality associated to the role.

DEFINITION **3.** *(Role). A role $r_i \in OR_i$ is defined by the tuple (ϕ_i, S_i) , where:*

- *ϕ_i is a semantic concept for the role;*
- *$S_{r_i} = \{s_1, \ldots, s_l\}$ is the set of semantic service descriptions associated to the role. Each service description is defined by the tuple $s_i = (I_i, O_i, Pr_i, E_i)$, where the components are ontological concepts grouped in: a set of inputs (I_i), outputs (O_i), preconditions (Pr_i), and effects (E_i) of the services, respectively.*

The structure of the system is defined by the relationships between agents. A relationship between two agents is established based on a social feature called *homophily* ($H(a_i, a_j)$) [27], which is considered to be self-organization criterion [12]. The homophily concept translated to the agent context has been considered as the similarity between two agents based on service and organizational role information. Two agents in the system have a high degree of homophily if they play similar roles and offer similar services. The degree of homophily (H) between two agents ranges in the interval [0,1], where 0 indicates that agents play different roles and offer different services and 1 indicates that agents play similar roles

Algorithm 1 Function that describes how an agent a_i starts a service discovery process.

```
1: function $startDiscovery()$
2: $\mathcal{R}_i \leftarrow \mathcal{R}_i + 1$
3: $q \leftarrow s_q, r_q$
4: $steps \rightarrow 0$    /* number of steps associated to the query */
5: $part \rightarrow \{\}$    /* list of participants in the service discovery process */
6: $\varepsilon \rightarrow 0.75$    /* threshold similarity */
7: $a_j, part \leftarrow serviceDiscovery(part, a_i, a_i, q, steps, \varepsilon)$
8: /* if a suitable provider agent is found before the TTL */
9: if $a_j \neq \emptyset$ then
10:     $request(a_j, s_q)$
11:     $\mathcal{P}_j \leftarrow \mathcal{P}_j + 1$
12:     for $a_n \in part$ do
13:         $\mathcal{SQ}_n \rightarrow \mathcal{SQ}_n + 1$
14:     end for
15: end if
16: end function
```

and offer similar services. Specifically, in the proposed system, agents establish links with other agents based on this homophily degree. Agents have a greater probability of establishing links with agents that have similar attributes than with dissimilar ones. The result of using this criterion to establish links between agents is a network structure based on homophily that has an exponential distribution of its degree of connection. This structure facilitates the task of decentralized service discovery only considering local information. For a detailed mathematical treatment about this process, we refer the reader to Appendix A.

4. Service Discovery Process

To illustrate the context where our proposal to facilitate the cooperation emergence is going to be applied, let us present the service discovery scenario where the service search process is described as well as the situations where the absence of cooperation affects the system performance.

A service discovery process relies on the cooperation of the agents. The process starts when an agent needs to locate an agent that plays a certain role and offers a certain service in order to deal with one of its goals. This process is described in Algorithms 1 and 2. Initially, agent a_i creates a query $q = \{s_q, r_q\}$, which consists of the required semantic service description and the organizational role that the target agent should play, and also increases by one the number of requests generated in the system, initializes the number of steps associated to the query to 0, and creates an empty list to store the identifiers of the agents that participate in the search (Alg. 1 Lines 2-5). The similarity threshold ε to consider that an agent offers the required service is established by the initiator agent a_i. With this information the service discovery starts (Alg. 1 Line 7).

The service discovery is carried out only if the number of steps of the query are lower than the Time To Live (TTL) and the query has not been solved. In the discovery process, when an agent that is similar enough to the target agent a_t is found (Alg. 2 Lines 5-7), the source agent that initiates the service discovery is informed and the process ends. Otherwise, agent a_i must choose one of its neighbors to forward the query (Alg. 2 Line 9). The *promisingNeighbor* function describes the process of selecting the neighbor that has the greatest probability of achieving the target in the fewest steps; it is based on the following function $\mathcal{F}_{N_i} : A \rightarrow N_i$:

$$\mathcal{F}_{N_i}(a_t) = argmax_{a_j \in N_i} \left[1 - \left(1 - \left(\frac{H(a_j, a_t)}{\sum\limits_{a_n \in N_i} H(a_n, a_t)} \right) \right)^{|N_j|} \right] \tag{1}$$

The parameter a_t is a *target* agent $a_t = (r_q, s_q, \emptyset, \emptyset)$ that represents the unknown provider agent that is able to provide both a service that is similar to the service that appears in the query and a role that is similar to the role that appears in the query. This formula uses a homophily-based factor (H) [12] and a degree-based factor (number of neighbors $|N_j|$) to select the most promising neighbor. The homophily-based factor considers the semantic similarity between the organizational roles and semantic services of agent a_j and the unknown provider agent a_t. The divisor of the expression is just a normalization factor.

If the neighbor accepts to forward the query, it increases by one both the number of queries forwarded and the number of steps of the query, and also adds its identifier (id) to the list of participants in the process and continues with the service discovery process (Alg. 2 Lines 9-13). As we have said before, this process continues until a suitable agent is found or the number of times the query has been forwarded is over the TTL. In the latter case, the last agent sends a report (inform message) to the agent that initiated the process with an empty list of participants and without a provider agent.

Finally, when the agent that started the discovery process receives the inform message, it analyzes the content. If the message contains an identifier of a provider agent, it means that the service discovery ends successfully. Therefore, the provider agent increases by one the number of requests attended to, and

Algorithm 2 Function that describes the service discovery process that an agent a_i carries out when it receives a query.

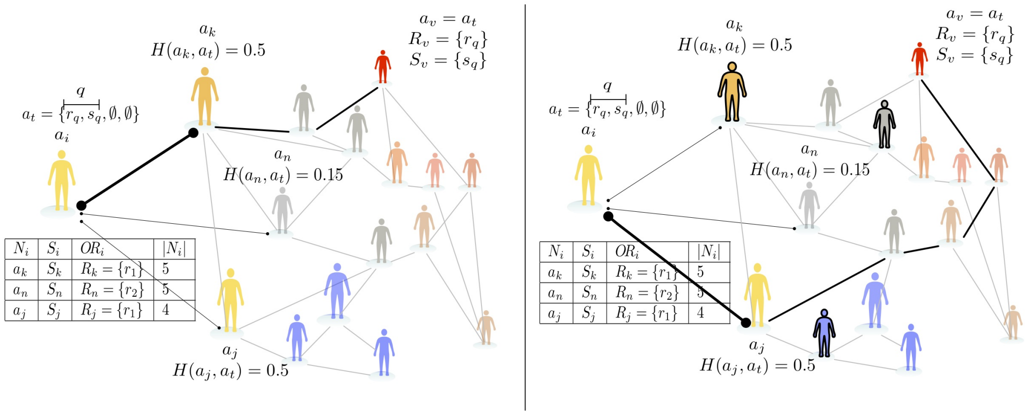
```
function $serviceDiscovery(part, a_{source}, a_i, q = (s_q, r_q), steps, \varepsilon)$
$found \leftarrow false$
$a_t \leftarrow (r_q, s_q, \emptyset, \emptyset)$     /* target agent that represents the profile of a suitable provider agent to solve the query */
if $steps \leq TTL$ then
    /* if an agent enough similar to the target is found */
    if $H(a_i, a_t) \geq \varepsilon$ then
        $found \leftarrow true$
        $inform(a_{source}, part, a_i)$
    else
        /* if the agent is not enough similar, it selects the most promising neighbor*/
        $a_i \leftarrow promisingNeighbor(a_i, a_t)$
        if $a_i \neq \emptyset$ then
            $\mathcal{Q}_i \rightarrow \mathcal{Q}_i + 1$
            $part \rightarrow part \cup \{a_i\}$
            $steps \leftarrow steps + 1$
            $serviceDiscovery(part, a_{source}, a_i, q = (s_q, r_q), steps, \varepsilon)$
        else
            $inform(a_{source}, \emptyset, \emptyset)$
        end if
    end if
else
    $inform(a_{source}, \emptyset, \emptyset)$
end if
end function
```

the agents that participate in the search process increases by one the number of queries forwarded that contribute to a successful discovery process (Alg. 1 Lines 9-14).

As an example, let's see the scenario in Figure 1a. In this scenario, all the agents are cooperative. Agent a_i generates a query q and it should choose one of its neighbors, a_n, a_j, or a_k, to forward the query. In order to select the most promising neighbor, the agent a_i applies Function 1. This function considers: (i) the homophily between the neighbors of agent a_i and a *target* agent $a_t = (s_q, r_q, \emptyset, \emptyset)$ that offers the service and plays the role specified in the query q; and (ii) the degree of connection of the neighbors. Assuming the values of homophily that appear in Figure 1a ($H(a_k, a_t) = H(a_j, a_t) = 0.5$, and $H(a_n, a_t) = 0.15$):

$$\begin{aligned}\mathcal{F}_N(a_t) &= argmax_{a_k,a_j,a_n}\left[1-\left(1-\frac{0.5}{1.15}\right)^5, 1-\left(1-\frac{0.5}{1.15}\right)^4, 1-\left(1-\frac{0.15}{1.15}\right)^5\right] \\ &= argmax_{a_k,a_j,a_n}[0.942, 0.897, 0.5] = a_k\end{aligned}$$

Therefore, agent a_i sends the query to the most promising agent (i.e., agent a_k). In that case, the forwarding process is repeated until an agent similar enough to a_t is found. In this scenario the service discovery process ends when agent a_v is



N_i	S_i	OR_i	$\lvert N_i \rvert$
a_k	S_k	$R_k = \{r_1\}$	5
a_n	S_n	$R_n = \{r_2\}$	5
a_j	S_j	$R_j = \{r_1\}$	4

(a)

N_i	S_i	OR_i	$\lvert N_i \rvert$
a_k	S_k	$R_k = \{r_1\}$	5
a_n	S_n	$R_n = \{r_2\}$	5
a_j	S_j	$R_j = \{r_1\}$	4

(b)

Figure 1: Service discovery scenario. Note that agents that play similar roles and offer similar services have been represented with similar colors. (a) All the agents are cooperative. Agent a_i only knows its direct neighbors a_k, a_j, and a_n. If a_i needs to locate a service (i.e., s_q), it will forward the query to its most promising neighbor (i.e., a_k) based on the homophily between the neighbor and the target agent (i.e., a_t) that should provide the required service and the degree of the neighbor. (b) Agents with thick silhouette are not cooperative. Agent a_i has to find an alternative agent to a_k since a_k refuses to forward queries.

found.

The scenario of Figure 1b is similar to scenario 1a but here there are agents, such as a_k (thick silhouette), that are not cooperative and decide not to forward queries from other neighbors compromising the service discovery process. Therefore, although agent a_i considers that the most suitable agent to forward the query is a_k, agent a_i should find another neighbor, a_j, to forward the query to that may not be as suitable as the first one but that has a higher degree of cooperation. As a result of this non-cooperative behavior, the number of steps required to reach the target agent a_v increases considerably.

In this particular scenario of service discovery, if the number of agents that decide not to forward queries from other agents increases, the length of the paths and the messages generated in the discovery process increases considerably since non-cooperative agents should be avoided and alternative paths should be found. Therefore, cooperative agents must forward more queries, the number of searches that end successfully is reduced, and the system efficiency decreases. The worst case is when the length of a path is near to the TTL and the service is finally not found. In this situation, the effort of all cooperative agents is useless. For that reason, it is important to provide mechanisms to be able to confront situations where agents are pursuing their own goals without cooperating and are compromising the performance of the overall system [25]. These mechanisms should promote and maintain cooperation in the system in order to guarantee the proper performance of the system [13].

In the following sections, we describe two mechanisms that facilitate the emergence of cooperation. One of them is based on local structural changes. The other one is based on the use of incentives. We also describe under which circumstances are more appropriate to use them. Finally, we explain how both mechanisms can be combined to improve the performance of the service discovery and maintain the degree of cooperation even in scenarios where the predominant behavior is not collaborate.

5. Structural Mechanism: Social Plasticity

Structure is an important feature to consider in the cooperation models [28]. The structure of the network influences interactions of agents, therefore it is important to provide agents mechanisms to be able of changing their local structure in the network. In this section we propose the inclusion of a structural mechanism. Through interactions during the service discovery process, agents are able to change their relations taking into account which neighbors provide profitable relationships and which do not. This feature is called *social plasticity* [14]. Social plasticity is the capacity of individuals to change their relationships as time passes. Specifically, in our system, each agent maintains information related to its neighbors. This information consists of the number of times a neighbor has refused to forward one of its queries $\mathcal{RQ}_{ij}$. The agent keeps a counter for each of its links in its internal state (st_i). Each counter is increased by one unit each time that a query is refused by a neighbor. If a neighbor decides to change its behavior and forwards queries, the agent updates the counter to 0. Therefore, a cooperative agent that occasionally cannot forward queries, because its workload at a certain moment is too high to attend to more queries is not considered to be a non-cooperative agent.

Each time an agent tries to forward a query to one of its neighbors a_j, it updates its information associated to the link with a_j and evaluates the utility of the link. In order to evaluate the utility of a link, an agent a_i uses a decay function that calculates the probability of maintaining a link with a_j taking into account the

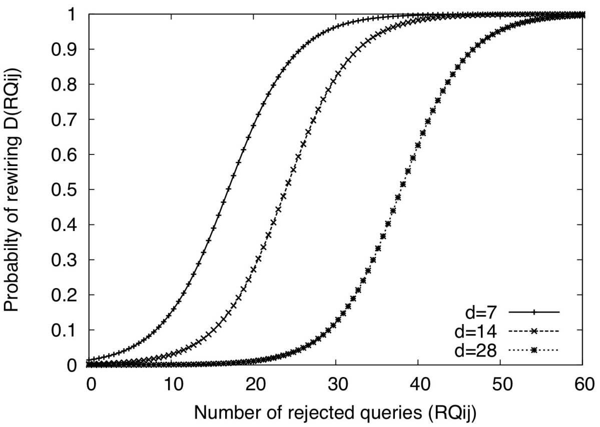


Figure 2: Sigmoid function that calculates the probability of breaking a current link with a neighbor and looking for a new neighbor. The figure shows the shape of the function with different values of displacement parameter d.

number of queries that it would have sent through neighbor a_j but that a_j refused to forward. This function is a sigmoid that ranges between [0,1],

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

where $\mathcal{RQ}_{ij}$ is the number of queries that neighbor a_j received from agent a_i and that a_j decided not to forward. The constant y is the slope and d is the displacement. These constants are established by the agent. The most influential constant is d. The displacement d indicates how benevolent an agent is with respect the non-cooperative behavior of its neighbors. A high value of d means that the agent is going to consider a higher number of refuses in order to make a decision about looking for another neighbor. A low value means that it is not permissive with the number of refuses (see Figure 2). The function $D(\mathcal{RQ}_{ij})$ returns a value in the range [0,1], where 0 indicates that the agent does not consider that the number of rejects from its neighbor is enough to make a decision about rewiring, and 1 indicates that it is necessary to change the link (see Figure 3).

Algorithm 3 shows the *socialPlasticity* function where an agent a_i evaluates its link with one of its neighbors a_j. The agent a_i uses the D function and considering this probability it decides to break its current link and look for a neighbor

Algorithm 3 Function that an agent uses to analyze if it is appropriate to change a link.

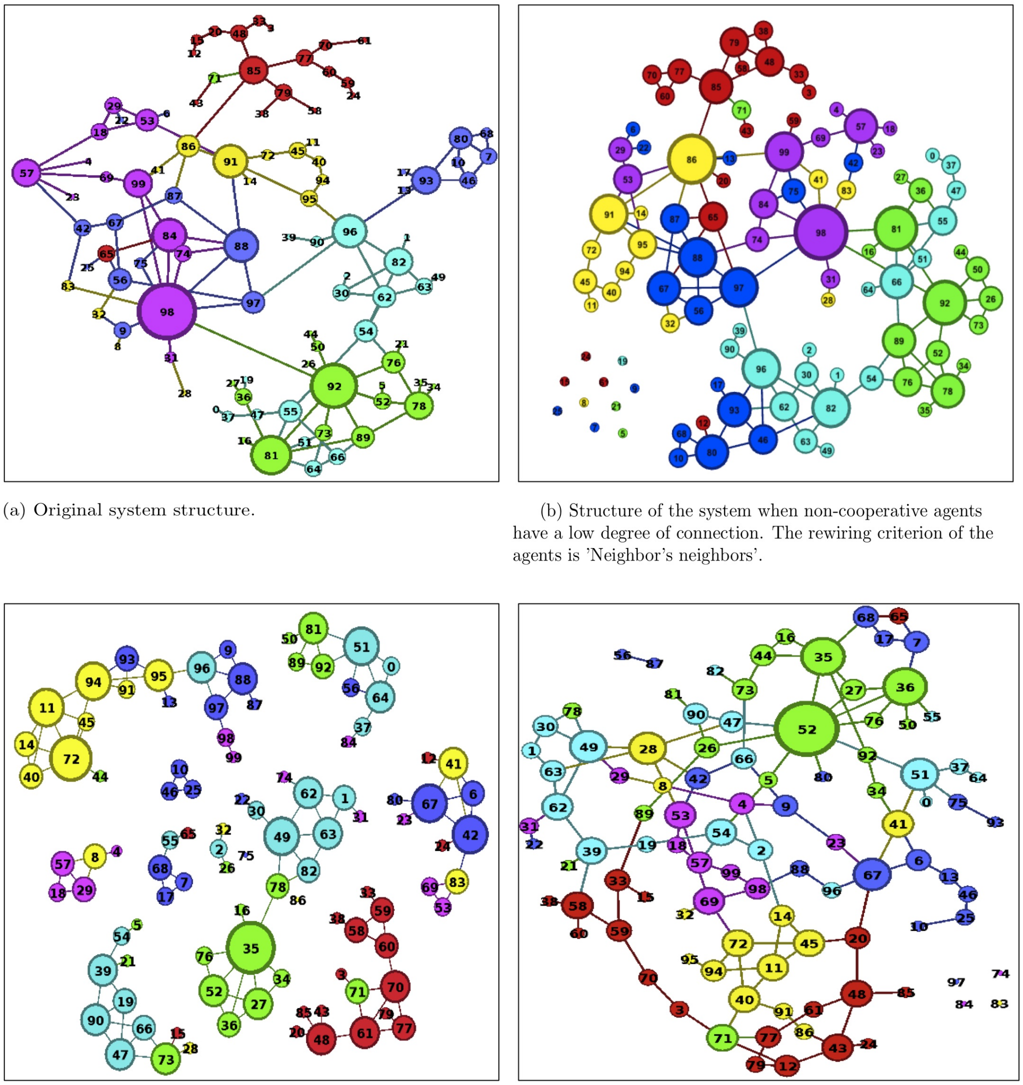
```
1: function $socialPlasticity(a_i, a_j)$
2: if $D(RQ_{ij}) > (random(0, 1))$ then
3:     $N_i \leftarrow N_i - \{a_j\}$
4:     $newAg \leftarrow lookForNeighbor(a_i)$
5:     $N_i \leftarrow N_i \bigcup \{newAg\}$
6:     $N_{newAg} \leftarrow N_{newAg} \bigcup a_i$
7: end if
8: end function
```

to maintain its degree of connectivity. In the case that agent a_i decides to break the link with neighbor a_j, agent a_i looks for another agent to establish a new link in order to maintain its degree of connectivity (Alg. 3 Line 4). We assume that any alternative agent (cooperator or non cooperator) always accepts a new partner. There are different criteria for establishing a new link with another agent in the network. We have considered two criteria:

- establishing a link with a *neighbor's neighbor* [14];
- looking for a *similar neighbor to the previous neighbor*.

To illustrate how these criteria affect the system structure, let us begin with simple examples of networks where agents have social plasticity to ostracize agents that are not cooperating. Figure 3 shows the effect of ostracizing non-cooperative agents with different rewiring mechanisms. Each network has 100 agents. Each agent is represented by a node and plays a role and offers a semantic service associated to the role. There are 6 roles defined in the system. Each color represents the role that an agent plays. The number inside a node is the identifier of the agent. Agents are distributed uniformly over the roles and services in the system. The average degree of connection of the network is 2.5. Figure 3a shows the initial structure of the system. Note that the effect of homophily makes agents establish a higher number of connections with similar agents than with dissimilar ones. The number of non-cooperative agents is 25. An agent that breaks a link with another agent looks for a new one. As stated above, we consider two criteria to replace a link with a non-cooperative neighbor. We analyzed the effects of each criterion after 1,000 queries that were uniformly generated by the agents:

Neighbor's neighbors. The effects of using the criteria *Neighbor's neighbors* are shown in Figure 3b. The non-cooperative agents are the agents with id-labels that range in the interval [0,25]. The nodes that are non-cooperative are isolated as a result of the local decisions of the agents. The structure of the network remains



(a) Original system structure.

(b) Structure of the system when non-cooperative agents have a low degree of connection. The rewiring criterion of the agents is 'Neighbor's neighbors'.

(c) Structure of the system when non-cooperative agents have a high degree of connection. The rewiring criterion of the agents is 'Neighbor's neighbors'.

(d) Structure of the system when non-cooperative agents have a high degree of connection. The rewiring criterion of the agents is 'choice homophily'.

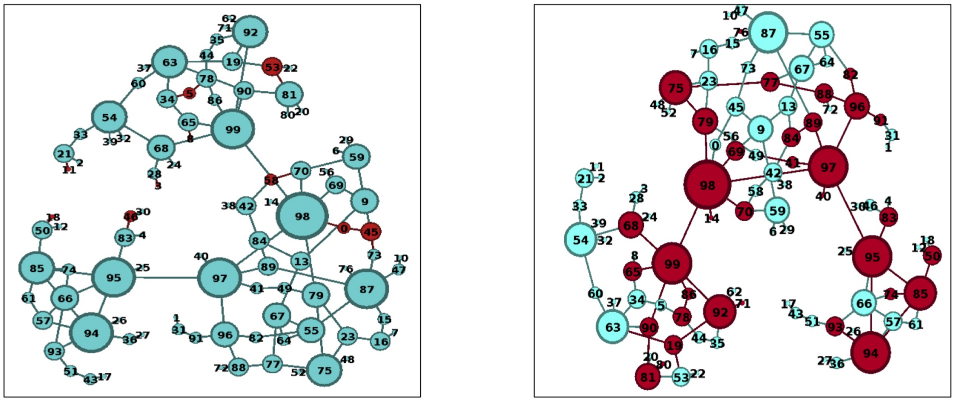
Figure 3: Effects of social plasticity on the structure of the system where 25 agents are non-cooperative and 75 cooperative.

connected and the success rate of the discovery process does not change. However, the structure of the network does not always remain connected. In this case, the non-cooperative agents have a low degree of connection and they do not are not located in a critical position in the structure of the network (i.e., hub that connects different communities of agents). A different situation is when non-cooperative agents have a high degree of connection. In Figure 3c, non-cooperative agents are those nodes with id-labels that range in the interval [99,74]. These agents with a high degree of connection play an important role in connecting different communities. The ostracism of these agents using the *Neighbor's neighbors* criterion not only disconnects or reduces the connectivity of non-cooperator agents but also divides the network into several isolated parts. On one hand, the rewiring strategy increases the degree of clustering inside the community. On the other hand, the rewiring strategy decreases the probability of establishing connections with different communities, thereby decreasing the success rate of the search process since services from different communities cannot be reached.

Similar neighbor to the previous neighbor. The effects of the mechanism based on the probability of establishing a link with an agent that offers similar services and plays similar roles to the previous neighbor are shown in Figure 3d. The probability of agent a_i establishing a connection with an agent similar to a_j is proportional to the homophily between the candidate neighbor and the current neighbor $(H(a_j, a_n))$. This strategy gives cooperative agents the opportunity to maintain their inter-community connections avoiding the creation of isolated communities. As Figure 3d shows, this mechanism allows agents to maintain the structure of the network even though non-cooperative agents are located on a critical position in the structure of the network. The non-cooperative agents (nodes with id-labels that range in the interval [99,74]) have lost all their connections or their degree of connection has been considerably reduced. Therefore, they lose their privileged position in the network and now are located on the fringes of the network or completely ostracized.

Therefore, we consider that the *Similar neighbor to the previous neighbor* criterion is suitable to avoid the system disconnection when social plasticity is considered by the agents and non-cooperative agents have a high degree of connection (i.e., act as hubs in the service discovery process). The maintenance of the connected system is important to maintain the success rate of the discovery process since services located in far positions could be reached.

The use of social plasticity to isolate or to reduce the degree of connection (thereby the influence of non-cooperative agents) improves the performance of



(a) Initially, the system was populated by a set of 25 non-cooperative agents and 75 cooperative agents. The degree of connection was uniformly distributed over agents. The figure shows the effects of using incentives to promote cooperation among agents.

(b) Initially, the system was populated by a set of 25 non-cooperative agents that had a high degree of connection (agents with ids in the range [99,75]). The figure shows the distribution of cooperation after the use of incentives to promote cooperation among agents.

Figure 4: Evolution of the number of cooperative and non-cooperative agents after 1,000 queries generated in the system. Initially, there were 75 agents that cooperate and 25 that did not cooperate. The nodes represent agents and the numbers are the identifiers of the agents. Blue nodes represent cooperative agents and red nodes non-cooperative agents.

the system. However, the use of structural mechanisms when the number of noncollaborator agents increases could break the network structure into several isolated parts, thus reducing the system performance. Moreover, it is difficult to break links with non-cooperative agents that are located in the fringes of the network and have a low degree of connection. These non-cooperative agents do not receive enough queries to consider a rewiring action. Note that, in scenarios where structural changes have a significant cost, this mechanism can not be always used. As an alternative to this method, in the following section, we propose an incentive mechanism that does not change the network structure to facilitate the emergence of cooperation.

6. Incentive Mechanism

In the proposed model, agents can have cooperative or non-cooperative behavior. Cooperating in the service discovery scenario implies that an agent is going to: forward queries, request services, and attend to requests 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 or a benefit. For instance, forwarding a request has a cost since an agent has to dedicate time and resources to decide which neighbor is the best one to forward the query to. If a query finally arrives to an agent that provides the required service (i.e., the search process ends before the TTL), then the agents that participated by cooperating in the forwarding process will obtain a benefit for their contribution. 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.

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 co-operators, cooperative behavior does not always emerge. In this case, the incentive of cooperating in the discovery process is not always enough to maintain cooperation.

An agent calculates its payoff with the following function based on its behavior and taking into account its actions:

$$\mathcal{PO}(st_i) = \mathcal{SQ}_i \cdot sq - \mathcal{Q}_i \cdot q + \mathcal{P}_i \cdot p - \mathcal{R}_i \cdot r \qquad (3)$$

where:

- $\mathcal{SQ}_i, \mathcal{Q}_i, \mathcal{P}_i, \mathcal{R}_i$ is the information of the internal state (st_i) of an agent (see Definition 2),
- q is the cost of forwarding queries,
- sq is the benefit obtained by the agents that participate by forwarding queries in a service discovery process that ends successfully,
- p is the benefit obtained by the agents that provide a service,
- r 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 update rule implemented in this model is based on imitation

Algorithm 4 Function that updates the behavior of an agent taking into account the payoff of its neighbors.

```
function $CooperationAnalysis(a_j)$
$payOff_j \leftarrow \mathcal{PO}(st_j)$
$neighborToImitate \leftarrow \emptyset$
for $a_n \in N_j$ do
    if $\mathcal{PO}(st_n) > payOff_j$ then
        $neighborToImitate \leftarrow a_n$
    end if
end for
$\mathcal{C}_j \leftarrow \mathcal{C}_{neighborToImitate}$
$dc_j \leftarrow dc_{neighborToImitate}$
return $\mathcal{C}_j, dc_j$
end function
```

[47]. 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 (see Algorithm 4).

We would like to remark that cooperative behavior analysis is asynchronous. Agents that analyze and update their cooperative behavior are those that during the service discovery process are considered to be the most promising candidates to forward a query, even though they finally do not forward the query.

To asses the impact of the incentive mechanism, we conducted several simulations in small networks of 100 agents and two different configurations. In the networks of Figure 4, 75% of the agents were cooperative and 25% were non-cooperative. The rest of the structural parameters of the networks were similar to the networks presented in Figure 3. The costs and benefits of the actions were: $q = 0.15$, $sq = 0.30$, $p = 0.5$, and $r = 0.5$. Agents update their behavior when they participate in the discovery process. In the networks of Figure 4, non-cooperative agents are represented by red nodes and cooperative agents by blue nodes.

Figure 4a shows the effects of the incentive mechanism after 1,000 queries generated in the system where the degree of connection is uniformly distributed over the agents without taking into account their behavior. In this scenario, the incentive mechanism is enough to promote cooperation among agents. The majority of agents that do not cooperate are situated on the fringes of the network since these positions are not easily influenced.

Figure 4b shows the effects of the incentive mechanism after 1,000 queries generated in the system where the non-cooperative agents had a high degree of connection. The non-cooperative agents got benefits quickly since they received a high number of service requests and they do not have the cost of forwarding others' queries. The cooperative agents had a great number of losses when agents with a high degree of connection did not cooperate because the discovery process took more steps, and, therefore, cooperative agents had the cost of forwarding queries but they had a low probability of receiving a benefit since the number of non-cooperators was too high and the probability of being successful decreased significantly. In this scenario, non-cooperative agents obtained a higher payoff than cooperative agents, and, therefore, had a greater influence on their neighborhood. Although the influence of the non-cooperative agents was clear, their influence was not enough to convert all the cooperative agents into non-cooperative agents. There are some special situations where cooperative agents have influence over the non-cooperative even though they have a low degree of connection. These cooperative agents are located on the fringes of the network with a degree of 1. Therefore, they have less probability of participating in the search process, and they do not have many losses because of the forwarding process. This fact gives them more benefits than their neighbors, and they can influence their behavior. Moreover, nodes that have a neighborhood with the same behavior and a low degree of connectivity do not change their behavior. However, this is not enough to influence nodes beyond the neighborhood.

7. Adaptive Combination of Social Plasticity and Incentives

The use of structural mechanisms such as social plasticity or incentives promotes the emergence of cooperation. Nevertheless, in scenarios where the predominant behavior is to not cooperate and non-cooperative agents have a significant influence (i.e., high degree of connection), the separate use of these mechanisms is not enough. Social plasticity could break the network into several isolated parts and if structural changes imply an economic cost, not all the agents may be able to afford them. A mechanism based on incentives is enough when the number of non-cooperative agents is low, but in other situations the expected payoff does not compensate the effort to cooperate. Therefore, we propose the integration of both mechanisms in order to find a trade-off between the costs and the degree of cooperation achieved in the network.

Algorithm 5 shows how these two mechanisms have been combined and integrated in the service discovery process. Specifically, both mechanisms are included in the process carried out by each agent when it has to decide the most promising neighbor to forward the query to the target agent. Once an agent a_i has selected a neighbor a_j, agent a_j evaluates its behavior comparing its payoff with the rest of its direct neighbors (Line 7). Based on this comparison, the agent de-

Algorithm 5 Function that describes how an agent selects the most promising neighbor when heterogeneous agents that cooperate and do not cooperate are considered.

```
1: function $promisingNeighbor(a_i, a_t)$
2: $N'_i \leftarrow N_i$        /* list of candidate neighbors to consider to forward the query */
3: $numCoop \leftarrow 0$       /* number of cooperative neighbors */
4: $coop \leftarrow false$
5: repeat
6:     $a_j \leftarrow \mathcal{F}_{N'_i}(a_t)$  /* Most promising neighbor selection function (see Formula 1) */
7:     $\mathcal{C}_j, dc_j \leftarrow cooperationAnalysis(a_j)$
8:     /* if the neighbor $a_j$ does not cooperate, $a_i$ considers the use of social plasticity */
9:     if $dc_j < random(0, 1)$ then
10:        $\mathcal{RQ}_{ij} \leftarrow \mathcal{RQ}_{ij} + 1$
11:        $numCoop \leftarrow coopNeighbors(a_i)$       /* number of cooperative neighbors of agent $a_i$*/
12:        /* if the number of cooperative neighbors of agent $a_i$ is under a threshold ($N_i \cdot \rho$)*/
13:        if $coop < N_i \cdot \rho$ then
14:            $socialPlasticity(a_i, a_j)$
15:        end if
16:        $N'_i \leftarrow N'_i - \{a_j\}$       /* the neighbor $a_j$ is deleted from the list of candidates to consider to forward the
           query */
17:        $coop \leftarrow false$
18:    else
19:        $coop \leftarrow true$
20:        $\mathcal{Q}_j \leftarrow \mathcal{Q}_j + 1$
21:    end if
22: until $\neg coop \vee (|N'_i| == 0)$       /* until a cooperator neighbor is found or the list of candidates is empty */
23: if $(|N'_i| == 0)$ then
24:    return $\emptyset$
25: else
26:    return $a_j$
27: end if
```

cides whether or not to change its behavior in order to improve its payoff in future interactions.

If agent a_j does not cooperate, then a_i increases the number of times its neighbor a_j has refused to forward a query. Moreover, a_i evaluates whether or not it is appropriate to rewire the current link with a_j in order to find a better connection. 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 agents that an agent has in its neighborhood (Lines 10-13). 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. If a_j does not cooperate, a_i eliminates a_j from the set of neighbors to consider in the current forwarding process and repeats the process until a cooperative neighbor is found or the set of neighbors to consider is empty (Lines 14 and 18).

Finally, if a_i finds a neighbor that cooperates, the neighbor increases by one its local information about the number of queries forwarded (Line 17). Otherwise, the search process fails.

With the combination of the two mechanisms, 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. In situations where the number of non-cooperative agents is significant, this mechanism allows the emergence of cooperation. Some agents remain non-cooperative because they are located where the degree of clustering and the degree of connection are too low; therefore, the number of services provided and the queries forwarded are too low to influence the others.

8. Experiments

In this section we evaluate the proposed mechanisms to promote cooperation in service-oriented environments. First, we evaluate the social plasticity mechanism. Specifically, we analyze the behavior of the mechanism with different configuration parameters and in different networks. Second, we test the behavior of the incentives mechanism when different reward values are used in different network configurations. Finally, we evaluate the combination of both mechanism and we compare it with other approaches used to promote cooperation in distributed environments.

The tests were performed on a set of 10 undirected networks based on preferences where the degree of connection followed and exponential distribution. 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 two features that characterize the required provider agent: the role and the service. 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 before the

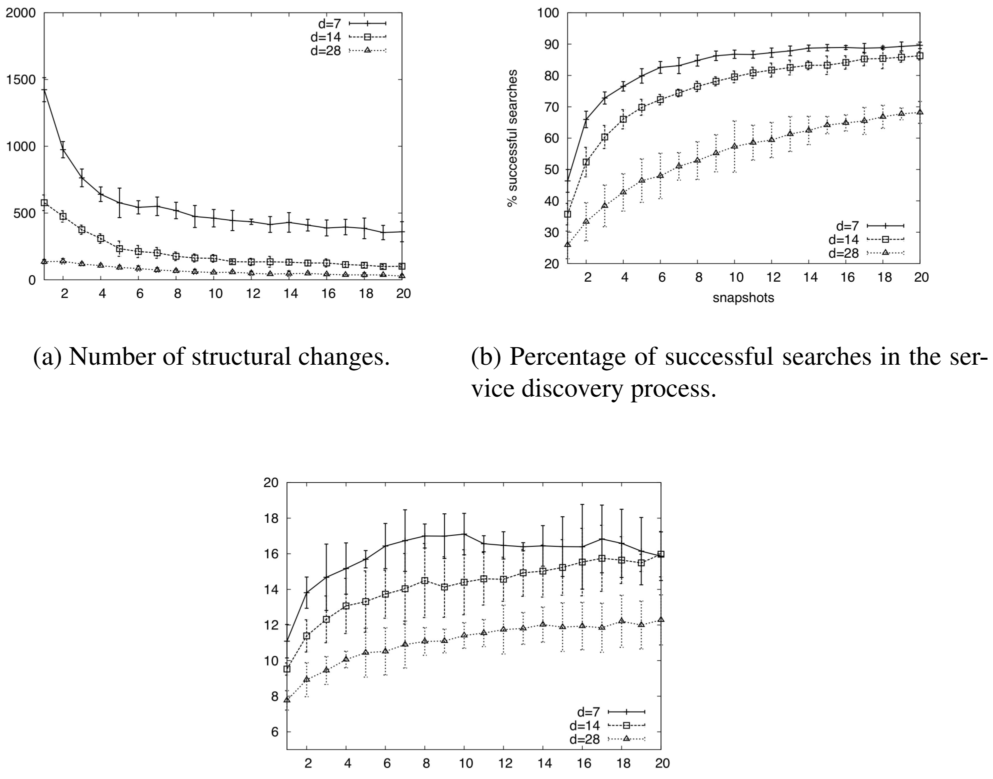
[1] http://www.semwebcentral.org/projects/owls-tc/

TTL ($TTL = 100$). The query distribution in the system was modeled as a uniform distribution. In the experiments, we made a snapshot of all of the metrics every time 5,000 queries were solved in the system in order to see the evolution of the metrics.

8.1. Social Plasticity

In order to see the influence of the displacement parameter d, we tested the behavior of the service discovery system when agents incorporate social plasticity with different values of d. The tests were done in networks where there were 600 non-cooperative agents and 400 cooperative agents. Figure 5a shows the total number of structural changes in the system in each snapshot. Each snapshot consisted of 5,000 queries. When agents were configured with $d = 7$, they were less benevolent with their neighbors and the number of structural changes was larger than in the other configurations. Note that in all configurations the highest number of changes were in the first iterations. The effect of a high number of structural changes is that the system isolates the number of non-cooperators earlier. This makes the success rate in configurations with low values of d ($d = 7$ and $d = 14$) increases earlier than with other configurations with high values ($d = 28$) (see Figure 5b). The average path length of successful searches increases as the non-cooperative agents are isolated (see Figure 5c). At the first iterations, the only searches that ended successfully were those that located a suitable provider agent in a few steps because, in short paths, there was a lower probability of finding a non-cooperator agent. In the following snapshots, the agents isolated the non-cooperative agents and the path length of successful searches increased. This happened because not only can queries about services provided by nearby agents be solved, but also queries about services that are provided by agents located outside of the circle of interaction of the agents that initiated the discovery process can be solved. Note that the configurations $d = 7$ and $d = 14$ obtained similar results of success and average path length. However, the configuration $d = 14$ made a lower number of structural changes than $d = 7$. Therefore, for scenarios where structural changes have a cost associated to them, the configuration of decay function $D(\mathcal{RQ}_{ij})$ for evaluating the links of an agent with $d = 14$ is better than $d = 7$.

In order to see the effects of the introduction of heterogeneous agents with different degrees of cooperation on the performance of the service discovery system, we conducted several simulations. Figure 6a shows the success rate of the queries. The x-axis shows the number of non-cooperative agents. The y-axis shows the % of searches in which the agent that started the service discovery process found the



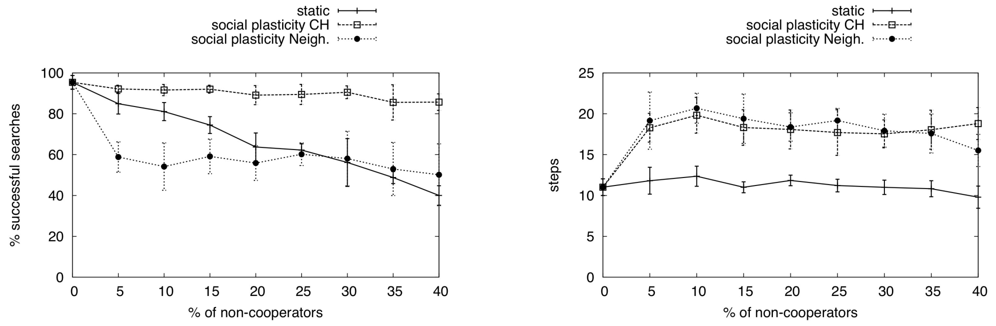
(a) Number of structural changes.

(b) Percentage of successful searches in the service discovery process.

(c) Average path length of successful searches.

Figure 5: Influence of displacement parameter d in a service discovery system where there are 600 non-cooperative agents and 400 cooperative agents. Each snapshot consists of 5,000 queries.

required service before the TTL. As expected, in systems with a static structure, where agents could not break links with non-cooperative agents, the percentage of success in the service discovery process decreased considerably. On the other hand, in dynamic systems, providing the agents with social plasticity to modify their relations benefits the search process. If agents isolate non-cooperative agents, the success rate decreases but not as significantly as in static networks. The search process in dynamic networks obtained a success rate of over 80% in the case where 400 agents were not collaborating in forwarding tasks. This was an increase of 40% over the success rate obtained in static networks. The success



(a) % of queries that end successfully.

(b) average number of steps in service discovery.

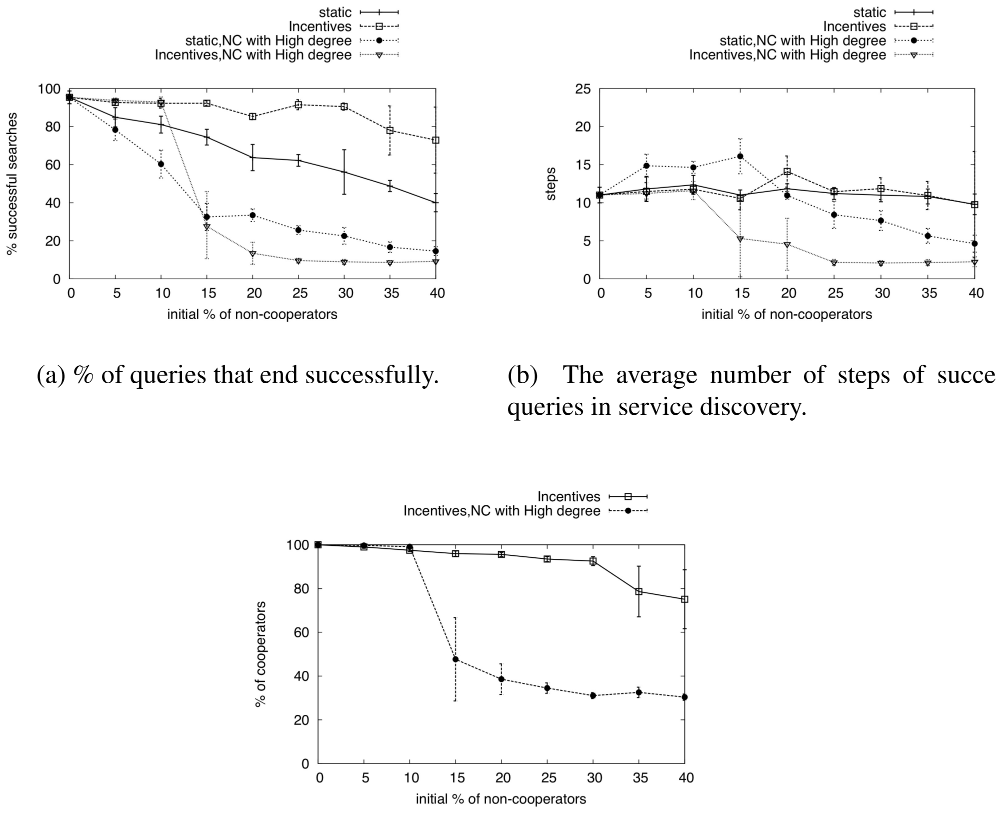
Figure 6: Performance evaluation as the number of non-cooperative agents increases.

in static systems decreases linearly with respect to the number of non-cooperative agents. We can conclude that the use of social plasticity to isolate non-cooperative agents benefits system performance.

Figure 6b shows how the path length of queries that were successfully solved changes as the number of non-cooperative agents increased. The x-axis shows the number of non-cooperative agents in the system. The y-axis shows the mean path length of the successful searches. The effect in static networks was an increase in the path length when the number of non-cooperative agents is around 50. Then, the path length decreased since the high number of non-cooperators made that only queries that could be solved by near agents ended successfully. Cooperative agents had to find alternative paths, which took more steps in order to avoid the non-cooperative agents. In the case of dynamic networks, the path length increased and remained around 18 steps. This is because agents were able to find alternative paths avoiding non-cooperative agents. Note that the error intervals are larger in dynamic systems than in the static systems, and, therefore, the variability in the path length. This is because as the number of isolated non-cooperators increases, the probability of significantly modifying the structure of the system increases, and therefore the path length increases.

8.2. Incentives

We analyzed the effects of the use of incentives in the success rate of the discovery process, in the average path length of successful searches, and in the



(a) % of queries that end successfully.

(b) The average number of steps of successful queries in service discovery.

(c) Evolution of the number of cooperator agents.

Figure 7: Performance evaluation of incentives mechanism as the number of non-cooperative agents increases in two types of networks: networks where all agents had the same probability of having a high degree of connection, and networks where non-cooperator had a high degree of connection.

	C	NC
C	(1, 1)	(0, 1.5)
NC	(1.5, 0)	(0, 0)

	C	NC
C	(0.15, 0.15)	(-0.15, 0)
NC	(-0.15, 0)	(0, 0)

Figure 8: Payoff matrix: (Left) Prisioner's Dilemma [14], (Right) Stag Hunt Dilemma.

emergence of cooperative behavior in the system. We considered two types of networks: networks where cooperative and non-cooperative agents had the same probability of having a high degree of connection, and networks where non-cooperative agents had a high degree of connection. In the experiments 75,000 queries were generated in 10 different networks of each type. The costs and benefits of the actions were: $q = 0.15$ (cost of forwarding action), $p = 0.5$ (benefit of providing a service), $r = 0.5$ (cost of asking for a service), and $sq = 0.30$ the reward of the forwarding action.

The results are shown in Figure 7. In networks where all the agents had equal probability of having a high degree of connection the incentives were enough to promote cooperation. The success rate was not seriously affected until the network had the 25% of the initial agents with a non-cooperative behavior. In the case of networks where non-cooperative agents had a high degree of connection, the incentives mechanism could not promote cooperation when the percentage of initial non-cooperator was over the 10%. The presence of a high number of non-cooperators made a decrease in the success rate and in the average mean path of the successful searches since only those queries that could be solved in the neighborhood of the agent that generated the query.

8.3. Social Plasticity and Incentives

Several tests were performed to evaluate the effects of the combination of social plasticity and incentives to facilitate the emergence of cooperation in a decentralized service discovery system. In the experiments, we made a snapshot of all of the metrics every time 5,000 queries were solved in the system in order to see the evolution of the metrics.

We compared our proposal of combining social plasticity and incentives with the separate use of both mechanisms. Moreover, we also compared the proposal with other mechanisms present in the literature. The set of approaches that we considered in the tests were the following:

- **Social plasticity (SP)**: agents only consider social plasticity to promote cooperation in the system. The value used for the displacement parameter in the decay function was $d = 7$ and the value used for the slope parameter was $y = 4$.

- **Incentives**: agents only consider incentives to facilitate the emergence of cooperation. The costs and benefits of the actions were: $q = 0.15$, $sq = 0.30$, $p = 0.5$, and $r = 0.5$.

- **Incentives and Social Plasticity (Incentives+SP)**: agents consider the combination of incentives and social plasticity to facilitate the emergence of cooperation. The costs and benefits of the actions were: $q = 0.15$, $sq = 0.30$, $p = 0.5$, and $r = 0.5$, and the value for the displacement parameter was $d = 7$ and the value used for the slope parameter was $y = 4$.

- **Reinforcement Learning (RL)**: Reinforcement learning has been used as an approach for solving decision-making problems in multi-agent systems. It allows agents to dynamically adapt to changes without requiring global knowledge. The reinforcement learning method used to promote cooperation was WPL. This algorithm is based on the following idea to achieve convergence: slow down learning when moving away from a stable policy and speed up when moving towards a stable policy [1]. This idea is similar to the Win or Lose Fast (WOLF) method [9], but the WPL method offers better performance than WOLF. Each agent a_i maintains two matrices, π_i and Q_i, with two dimensions $|st| \times |acc|$ (states and actions). There are two possible states in the context of cooperation: cooperate or not cooperate. There are two possible actions considered in the context of service discovery are: forward or refuse to forward a query. The matrix π_i stores the probability of cooperating. The value $\pi_i(st, acc)$ represents the degree of cooperation of an agent in a certain state. The value $Q_i(st, acc)$ represents the expected reward an agent will obtain by executing an action when it is in a certain state; this value is based on the number of agents in the neighborhood of agent a_i that collaborated and the number of queries that agent a_i sent and were solved successfully. Moreover, each time an agent participates in the service discovery process it considers the re-organization of its links (add or remove links) based on its degree of connection and the initial average degree of connection of the network.

- **Game Theory.** We considered two type of games: the Prisioner's Dilemma

(**PD**) [14], where individuals might not cooperate even though it seems to be their best interest to do so; and Stag and Hunt (**SH**) [24], which describes a conflict between safety and social cooperation. The main difference between them is the payoff matrix (see Figure 8). In these games, cooperate implies forwarding queries and not cooperate rejecting forward queries. These games do not consider the actions of requesting services or providing services.

An agent a_i plays the PD (or the SH) with its neighbors (a_j) when it is considered as the most appropriate agent to forward the query. The process of updating the behavior of the agents is similar to the process described in [14]. Agent a_i and each agent in its neighborhood ($a_j \in N_i$) calculate their aggregate payoff within their neighborhood. Agent a_i updates its current strategy by imitating the strategy of the neighbor with the largest payoff. If a_i imitates a non-cooperative agent a_j, it breaks its link with a_j and establishes a new link with another agent taking into account the homophily criterion with a probability of $p = 0.01$.

To evaluate the 'Incentives+SP' mechanism and compare it with the other approaches, we considered two scenarios: (i) one in which the number of cooperators was greater than the number of non-cooperators, and (ii) one in which the number of non-cooperators was greater than the number of cooperators. Specifically, the tests in each scenario focus on a set of metrics that are meaningful for the analysis of the performance of the system: (i) the evolution of the number of cooperator agents in the system; (ii) the average number of steps required to locate an appropriate agent that solves a query; (iii) the percentage of queries that are solved before the TTL; (iv) the number of failures caused by the presence of non-cooperator agents.

8.3.1. Scenario with 600 cooperators and 400 non-cooperators.

In scenarios where the number of cooperator agents is greater than the number non-cooperator agents, the majority of the mechanisms improve the degree of cooperation in the system (see Figure 9 (Left)). The best degree of cooperation was achieved by the mechanism based on 'SH' and the mechanism based on 'Incentives'. The 'SH' mechanism achieved a high degree of cooperation since the initial degree of cooperation made agents to prefer risk and cooperate than safety and not cooperate. 'Incentives' and 'Incentives+SP' also achieved a high degree of cooperation since in this scenario the incentive for the forwarding action was worth it in the majority of service discovery process. However, the 'Incentives+SP' achieved

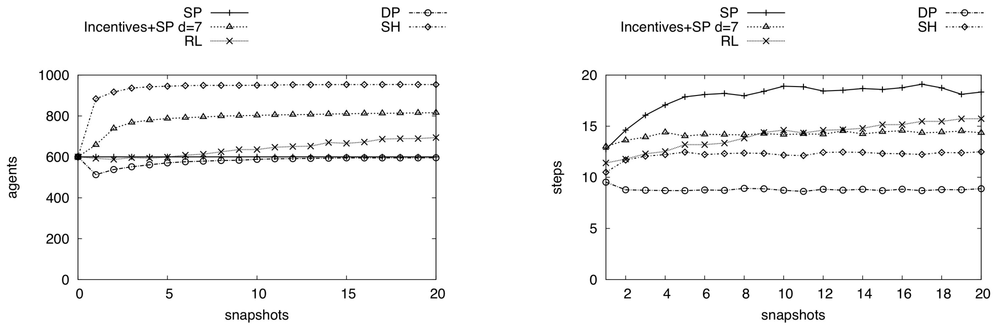


Figure 9: Evaluation of (Left) the evolution of cooperation in the system, and (Right) the average path length in the discovery process when there are 600 cooperative agents and 400 non-cooperative agents.

a lower degree of cooperation than ’SH’ due to the social plasticity isolated a high number of non-cooperator agents that could not change their behavior.

The average number of steps of queries that were successfully solved is shown in Figure 9 (Right). The largest average path length appeared with the ’SP’ mechanism. This is because it took more time to isolate the non-cooperative agents, and, therefore, the forwarding process had to avoid non-cooperative agents and required more steps. Moreover, with ’SP’ the number of potential provider agents is reduced by the ostracism of the non-cooperators. The mechanism ’Incentives+SP’ offered a lower number of steps than ’SP’ since the influence of incentives facilitated changes in the agent’s behavior and promoted cooperation. The ’SH’ and ’Incentives’ mechanisms offered shorter paths than ’Incentives+SP’ because the degree of cooperation achieved was nearly 100%; therefore, the number of queries that could only be solved by agents located far away with this mechanisms could be solved and the success in the discovery process increased. The ’PD’ mechanism obtained shorter paths than the other mechanisms since the number of non-cooperative agents did not decrease and this made that the only successful searches were those that could be solved by a nearby agent. Therefore, the average path length of successful searches was the shortest.

Figure 10 (Left) shows the number of searches that failed because to an agent could not find a neighbor that forwarded a request. The mechanisms that reduced the number of failures for non-cooperation were ’SP’ and ’Incentives+SP’. This is because both mechanisms isolated the non-cooperators quickly. ’Incentives’ and ’SH’ reduced the initial percentage of failures in the first snapshots and then re-

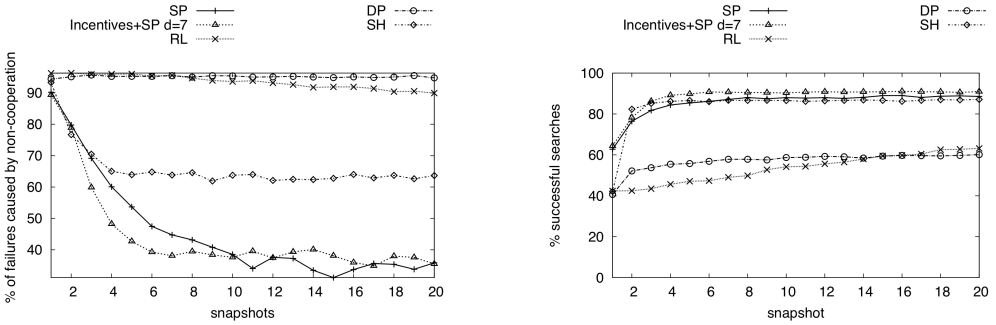


Figure 10: Evaluation of (Left) the percentage of failures because of the absence of cooperation, and (Right) the percentage of searches that end successfully in the system when there are 600 cooperative agents and 400 non-cooperative agents.

mained constant because the number of non-cooperators are located in the fringes of the network or in an strategic position that makes difficult change their behavior; therefore, the number of failures for the absence of cooperation remained constant. When the ’RL’ and the ’DP’ mechanisms were used, the number of non-cooperators could not be sufficiently reduced; therefore, the main reason for failure was non-cooperation.

Figure 10 (Right) shows the percentage of searches that were solved successfully. The strategies of ’SP’, ’Incentives’, ’Incentives+SP’, and ’SH’ obtained good results. The main difference among them is the number of iterations required to achieve a successful rate near 90%. The mechanism that offered the best results was the ’Incentives+SP’ because when the degree of cooperation in the local neighborhood increases, agents only consider incentives and do not use social plasticity; therefore, the number of agents that remain connected and can be considered to participate in the discovery process is higher than with other mechanisms such as ’SP’.

In this scenario, there are not significant differences between the mechanisms. In general, ’SP’, ’Incentives+SP’, and ’SH’ offer the best results. However, the ’SP’ and ’Incentives+SP’ are able to reduce the number of failures for not cooperating in the discovery process. The main differences between ’SP’ and ’Incentives+SP’ are the average path length since the ’SP’ reduces the number of potential provider agents and also the number of structural changes to maintain the proper performance of the system. In Figure 11, we compare the number of structural changes made by the agents using the ’SP’ and the ’Incentives+SP’

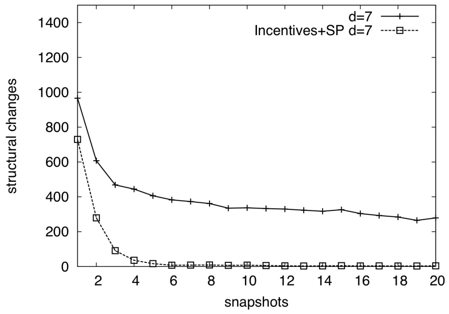


Figure 11: Number of structural changes in the system because of the social plasticity of the agents when there are 600 cooperative agents and 400 non-cooperative agents.

mechanisms. The combination of the two mechanisms reduces the number of structural changes required to achieve cooperation. This is important because there are scenarios where changes in the structure entail a cost and also because the excessive use of 'SP' reduces the number of potential provider agents to consider, and therefore, reduces the possibilities of finding a suitable provider in the service discovery process.

8.3.2. Scenario 400 cooperators and 600 non-cooperators.

In scenarios where the number of non-cooperators is greater than the number of cooperators, the mechanisms to facilitate the emergence the cooperation become more important. The behavior of the system when 600 non-cooperator and 400 cooperator agents are present in the system is evaluated. In this scenario, the differences between the mechanisms are greater than in the previous scenario since there is a greater number of non-cooperator agents.

Figure 12 (Left) shows the evolution of cooperation in the system when different mechanisms were used by the agents to promote cooperation. The best results were obtained by the 'Incentives+SP' mechanism. 'Incentives+SP' achieved the cooperation of the majority of agents in 5 snapshots. The 'SH' mechanism obtained worse results than 'Incentives+SP' mechanism due to the presence of a high number of non-cooperators. Agents prefer the safety of not cooperating over the risk of cooperating. The 'RL' mechanism maintained the cooperation level, but it could not increase it. The 'DP' mechanism did not promote cooperation

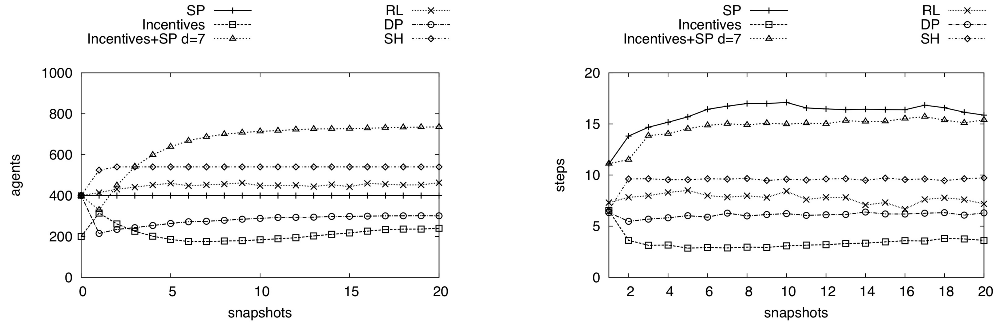


Figure 12: Evaluation of (Left) the evolution of cooperation in the system, and (Right) the average path length in the discovery process when there are 400 cooperative agents and 600 non-cooperative agents.

due to the payoff for not cooperating taking into account that the majority of the network did not cooperate. Therefore, the agents imitated the behavior of the agents with the highest payoff which were the non-cooperators. The 'Incentives' mechanism did not promote cooperation since the number of agents that did not cooperate forced the cooperators to invest resources in searches that were going to fail; therefore, the payoff of the cooperators decreased and the non-cooperators did not imitate them.

Figure 12 (Right) shows the average path length of successful searches. The 'SP' and 'Incentives+SP' mechanisms obtained longer paths because the number of potential provider agents was reduced since some of them could not be reached because they were isolated. Consequently, there were some service provider agents that took more steps to find. In the rest of the mechanisms, the number of non-cooperator agents was high enough to make the majority of the searches fail. Hence, the number of queries that ended successfully was low and these queries were those that could be solved near the neighborhood of the agent that generated the query.

Figure 13 (Left) shows the percentage of failures caused by non-cooperator agents. The 'Incentives+SP' and 'SP' mechanisms reduced the number of failures considerably. Since the other mechanisms could not deal with non-cooperators, the reason for the unsuccessful searches was the absence of cooperation. Figure 13 (Right) shows the percentage of successful searches. When agents used the 'Incentives+SP' or 'SP' mechanisms, cooperator agents were able to deal gradually with non-cooperators and improved the successful rate of searches. The

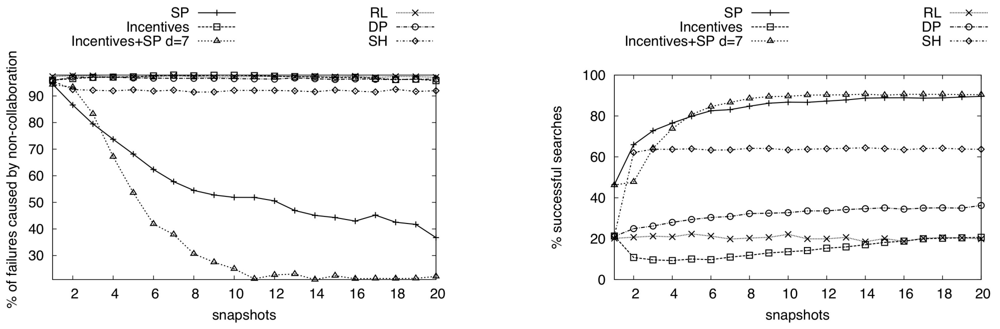


Figure 13: Evaluation of (Left) the percentage of failures because of the absence of cooperation, and (Right) the percentage of searches that end successfully in the system when there are 400 cooperative agents and 600 non-cooperative agents.

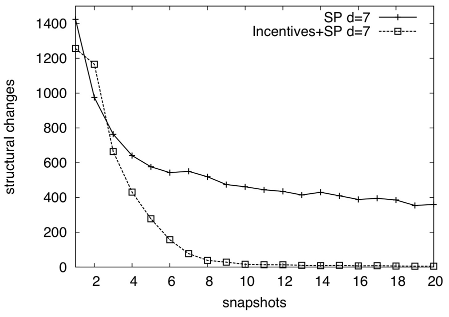


Figure 14: Number of structural changes in the system because of the social plasticity of the agents when there are 400 cooperative agents and 600 non-cooperative agents.

’SH’ mechanism also improved the success rate, but the improvement was not as significant as the improvement achieved by ’Incentives+SP’ or ’SP’.

Finally, Figure 14 compares the number of structural changes made by ’Incentives+SP’ and ’SP’. As in the previous scenario, agents were able to detect when it was necessary the use of incentives and social plasticity and when the use of incentives was enough to promote cooperation. Therefore, the number of structural changes was only used when the presence of non-cooperator agents was significant and this reduced the number of structural changes required to achieve a high degree of cooperation in the system.

9. Conclusions

This article addresses the problem of emergence of cooperation in scenarios where cooperation is required to achieve a good performance that benefits all of the participants. Specifically, our proposal focuses on the emergence of cooperation in decentralized service discovery scenarios where agents need the cooperation of their neighbors in order to locate other agents that offer services that they require. Therefore, if selfish agents appear in the system, in the long term, as the number of non-cooperator agents increases, the service discovery process could be seriously compromised. For this reason, it is important to provide mechanisms that facilitate the emergence and maintenance of cooperation. In this paper, we present the combination of two mechanisms to facilitate the emergence of cooperation in open service-oriented multi-agent systems where not all the agents have cooperative behavior.

On one hand, agents have social plasticity to change their structural relations based on the degree of cooperation of their neighbors. As the number of times a neighbor refuses to forward a query increases, the probability of changing this relation increases. If an agent decides to change a neighbor, it chooses a neighbor with similar functional features to the previous one. This avoids a high degree of fragmentation of the network when agents that act as hubs (i.e., they have a high number of connections) and non-cooperative agents are isolated by cooperative ones. However, this mechanism may not be appropriate if there are many non-cooperative agents since cooperative agents that are trying to isolate non-cooperative ones could change considerably the structure of the network. Moreover, if many agents are ostracized, the number of potential provider agents that could solve the query is reduced, and, therefore, the probability of success in the service discovery process is reduced. Furthermore, there are scenarios where breaking links could imply a cost; therefore, not all agent would be able to make use of social plasticity.

On the other hand, we propose the inclusion of incentives in order to influence the behavior of other agents. This mechanism associates a cost or a benefit to each action that an agent can carry out in the service discovery scenario. Forwarding queries have a benefit that depends on the success of the discovery process. Therefore, if the system is populated by a high number of cooperators, forwarding queries is considered to be a beneficial action with high probability. However, as the number of non-cooperators increases, this action becomes reckless and less profitable since the probability of finding an agent that has a neighborhood of non-cooperators increases and that ensures that the discovery process fails. Incentives are more appropriate for the emergence of cooperation in scenarios where the number of non-cooperators is not greater than the number of cooperators.

In order to deal efficiently with the emergence of cooperation even in scenarios where the number of non-cooperators is higher than the number of cooperators, we have proposed an adaptive combination of social plasticity and incentives. Agents considering local information are able to analyze and change their behavior, influence their neighbors, and decide when it is more appropriate the use social plasticity and incentives mechanisms or when it is enough with the incentives mechanism. With this combination, agents reduce the number of structural changes thereby avoiding the fragmentation of the network and the decrease of potential providers that can be considered during the service discovery process. The structural changes are enough to isolate non-cooperative agents and to increase the effectiveness of incentives in the emergence of cooperation even in scenarios where the majority of agents are non-cooperative. The experiments confirm that this combination of mechanisms promote cooperation in service discovery scenarios with different degrees of cooperation in the population of agents and offer better results than their use separate and than other approaches proposed for promoting cooperation in networks and that are based on game theory or reinforcement learning.

10. Acknowledgements

Work partially supported by the Spanish Ministry of Science and Innovation through grants TIN2009-13839-C03-01, CSD2007-0022 (CONSOLIDER-INGENIO 2010), FPU grant AP-2008-00601 awarded to E. del Val.

Appendix A. Homophily Calculation based on Semantics

The structure of the system is defined by the relationships between agents. A relationship between two agents is established based on a social feature called *homophily* ($H(a_i, a_j)$) [27], which is considered to be self-organization criterion [12]. The homophily concept translated to the agent context has been considered as the similarity between two agents based on service (H_v) and organizational role (H_s) information.

The similarity that takes into account the semantic service descriptions (H_v) is calculated between two sets of services, where $\mathcal{S}_i$ and $\mathcal{S}_j$ are the sets of services provided by the agents a_i and a_j, respectively. We consider each set of services $\mathcal{S}_i$ (or $\mathcal{S}_j$) to be composed by a set of semantic concepts that can be classified in: Inputs (I_i), Outputs (O_i), Preconditions (P_i), and Effects (Eff_i).

The level of matching between two sets of semantic concepts is calculated through a *weighted bipartite matching graph*. Let $G = (C_i, C_j, E)$ be a complete, weighted bipartite graph that links each concept $c_i \in C_i$ to each concept $c_j \in C_j$, $(c_i, c_j) \in E$, and let E represent the edges established in the graph $E = C_i \times C_j$. The term ω_{ij} represents the weight associated to the arc $e_i = (c_i, c_j) \in E$ between c_i and c_j as the semantic similarity between those concepts. For simplicity, we have considered four degrees of match (i.e., *exact, subsumes, plug-in*, and *fail* [39]) but other degrees could be considered [26]. The best match among concepts is obtained by calculating the *maximum weighted bipartite matching graph* $G' = (C_i, C_j, E')$, where $E' \subseteq E$ are the edges that have the maximal value. The graph G' is a relaxed bipartite graph because not all the concepts from C_j have to be connected to a concept in C_i; therefore, two concepts from C_i can share a concept from C_j. The weight of this graph is calculated as follows:

$$W_{G'} = \frac{\displaystyle\sum_{\omega_{ij} \in E'} \omega_{ij}}{\max\left(|C_i|, |C_j|\right)}. \tag{A.1}$$

Specifically, to calculate the H_v 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 ($\mathit{Eff}_i, \mathit{Eff}_j$)). The linear combination of the $W_{G'}$ of each set of concepts gives the value of the homophily between agents (see Equation A.2, where the parameters α and β assign different weights to the components of the formula).

$$\begin{aligned} H_v(S_i, S_j) = \alpha\big[\beta * W_{G'_I} + (1-\beta) W_{G'_O}\big] + \\ (1-\alpha)\big[\beta * W_{G'_P} + (1-\beta) W_{G'_{\mathit{Eff}}}\big] \end{aligned} \tag{A.2}$$

The similarity between the organizational roles (H_s) is calculated as the best match between the set of roles OR_i and OR_i played by the agents a_i and a_j. The match between two individual roles $r_i \in \mathit{OR}_i$ and $r_j \in \mathit{OR}_i$ is based on the distance between the semantic concepts ϕ_i and ϕ_j. The function presented by [17] is used to calculate the distance.

$$H_s(OR_i, OR_j) = \max_{r_i \in OR_i, r_j \in OR_j} rmatch(\phi_i, \phi_j)$$

where

$$rmatch(\phi_i, \phi_j) = \begin{cases} 1 & \text{if path length = 0} \\ \delta_{\phi_i \phi_j} \cdot e^{-\lambda(pl_{\phi_i \phi_j} + cp_{\phi_i \phi_j})} & \text{if } \phi_i \text{ and } \phi_j \text{ are not siblings} \\ \delta_{\phi_i \phi_j} \cdot e^{-\lambda(pl_{\phi_i \phi_j} + cp_{\phi_i \phi_j}) - d_{\phi_i \phi_j}} & \text{if } \phi_i \text{ and } \phi_j \text{ are siblings} \end{cases}$$

and

$$\delta_{\phi_i \phi_j} = \frac{e^{\gamma dp_{\phi_i \phi_j}} - e^{-\gamma dp_{\phi_i \phi_j}}}{e^{\gamma dp_{\phi_i \phi_j}} + e^{-\gamma dp_{\phi_i \phi_j}}}$$

The homophily $H(a_i, a_j)$ between two agents is calculated as a linear combination of the similarity between the services $H_v(S_i, S_j)$ and the similarity organizational roles of the agents $H_s(OR_i, OR_j)$. For a detailed mathematical treatment about this process, we refer the reader to [12].

References

[1] Abdallah, S., Lesser, V.: Multiagent Reinforcement Learning and Self-Organization in a Network of Agents. In: Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems, pp. 172–179. IFAAMAS (2007)

[2] Axelrod, R.M.: The evolution of cooperation. Basic Books, New York (1984)

[3] Basters, U., Klusch, M.: Rs2d: Fast adaptive search for semantic web services in unstructured p2p networks. In: Proceedings of the International Semantic Web Conference, *Lecture Notes in Computer Science*, vol. 4273, pp. 87–100. Springer (2006)

[4] Berendt, B., Hotho, A., Stumme, G.: Semantic web mining and the representation, analysis, and evolution of web space. In: Proceedings of the 1st International Workshop on Representation and Analysis of Web Space, pp. 1–16. Technical University of Ostrava (2005)

[5] Beydoun, G., Low, G., Tran, N., Bogg, P.: Development of a peer-to-peer information sharing system using ontologies. Expert Syst. Appl. **38**(8), 9352–9364 (2011)

[6] Bianchini, D., Antonellis, V.D., Melchiori, M.: Service-based semantic search in p2p systems. Proceedings of the European Conference on Web Services **0**, 7–16 (2009)

[7] Blake, M.B., Cummings, D.J., Bansal, A., Kona Bansal, S.: Workflow composition of service level agreements for web services. Decis. Support Syst. **53**(1), 234–244 (2012)

[8] Blanc, A., Liu, Y.K., Vahdat, A.: Designing incentives for peer-to-peer routing. In: Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies., vol. 1, pp. 374–385 vol. 1 (2005)

[9] Bowling, M., Veloso, M.: Multiagent learning using a variable learning rate. Artificial Intelligence **136**, 215–250 (2002)

[10] Crespo, A., Garcia-Molina, H.: Routing Indices For Peer-to-Peer Systems. In: Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS'02), p. 23. IEEE Computer Society (2002)

[11] Del Val, E., Rebollo, M., Botti, V.: Emergence of Cooperation through Structural Changes and Incentives in Service-Oriented MAS. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (2012)

[12] Del Val, E., Rebollo, M., Botti, V.: Self-Organized Service Management in Social Systems. In: Proceedings of the 45 Hawaii International Conference On System Sciences, pp. 810–817 (2012)

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

[14] Eguluz, V.M., Zimmermann, M.G., Cela-Conde, C.J., Miguel, M.S.: Cooperation and emergence of role differentiation in the dynamics of social networks. American Journal of Sociology **110**, 977 (2005)

[15] Ferber, J., Gutknecht, O., Michel, F.: From agents to organizations: An organizational view of multi-agent systems. In: Proceedings of the 4th International Workshop Agent-Oriented Software Engineering, *Lecture Notes in Computer Science*, vol. 2935, pp. 214–230. Springer (2003)

[16] Forestiero, A., Mastroianni, C., Meo, M.: Self-chord: A bio-inspired algorithm for structured p2p systems. In: Proceedings of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, pp. 44–51 (2009)

[17] Fu, P., Liu, S., Yang, H., Gu, L.: Matching algorithm of web services based on semantic distance. In: Proceedings of 2009 International Workshop on Information Security and Application (2009)

[18] Griffiths, N., Luck, M.: Changing neighbours: improving tag-based cooperation. In: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1 - Volume 1, pp. 249–256 (2010)

[19] Gu, B., Jarvenpaa, S.: 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)

[20] Hales, D., Edmonds, B.: Applying a socially inspired technique (tags) to improve cooperation in p2p networks. IEEE Trans. Systems, Man, and Cybernetics Part A: Systems and Humans pp. 385–395 (2005)

[21] Hardin, G.: The tragedy of the commons. Science (162), 1243–1248 (1968)

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

[23] Hofmann, L.M., Chakraborty, N., Sycara, K.: The evolution of cooperation in self-interested agent societies: a critical study. In: Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 2, pp. 685–692 (2011)

[24] Kamhoua, C., Pissinou, N., Makki, K.: Game theoretic modeling and evolution of trust in autonomous multi-hop networks: Application to network security and privacy. In: Proceedings of the IEEE International Conference on Communications (ICC), pp. 1–6 (2011)

[25] Kleinberg, J.: Complex networks and decentralized search algorithms. In: Proceedings of the International Congress of Mathematicians (ICM) (2006)

[26] Klusch, M., Fries, B., Sycara, K.: Owls-mx: A hybrid semantic web service matchmaker for owl-s services. Web Semantics Science Services and Agents on the World Wide Web **7**(2), 121–133 (2009)

[27] Lazarsfeld, P.: Friendship as a social process: A substantive and methodological analysis. Freedom and Control in Modern Society (1954)

[28] Lieberman, E., Hauert, C., Nowak, M.A.: Evolutionary dynamics on graphs. Nature **433**(7023), 312–316 (2005)

[29] Lin, W., Zhao, H., Liu, K.: Incentive cooperation strategies for peer-to-peer live multimedia streaming social networks. IEEE Transactions on Multimedia **11**(3), 396–412 (2009)

[30] Loia, V., Pedrycz, W., Senatore, S.: Semantic web content analysis: A study in proximity-based collaborative clustering. IEEE Transactions on Fuzzy Systems **15**(6), 1294–1312 (2007)

[31] Lopes, A.L., Botelho, L.M.: Improving multi-agent based resource coordination in peer-to-peer networks. Journal of Networks **3**, 38–47 (2008)

[32] Lv, Q., Cao, P., Cohen, E., Li, K., Shenker, S.: Search and replication in unstructured peer-to-peer networks. In: Proceedings of the 16th international conference on Supercomputing, pp. 84–95. ACM (2002)

[33] Michlmayr, E.: Ant algorithms for search in unstructured peer-to-peer networks. In: Proceedings of the 22nd International Conference on Data Engineering (ICDE) (2006)

[34] M.Luck, P.McBurney: Computing as interaction: Agent and agreement technologies. Proceedings of the IEEE SMC Conf. Distributed Human-Machine Systems (2008)

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

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

[37] Ohtsuki, H., Hauert, C., Lieberman, E., Nowak, M.A.: A simple rule for the evolution of cooperation on graphs and social networks. Nature **441**(7092), 502–505 (2006)

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

[39] Paolucci, M., Kawamura, T., Payne, T.R., Sycara, K.: Semantic matching of web services capabilities (2002)

[40] Pujol, J.M., Delgado, J., Sangüesa, R., Flache, A.: The role of clustering on the emergence of efficient social conventions. In: IJCAI, pp. 965–970 (2005)

[41] Santos, F.C., Santos, M.D., Pacheco, J.M.: Social diversity promotes the emergence of cooperation in public goods games. Nature **454**(7201), 213–216 (2008)

[42] Shen, J., Beydoun, G., Yuan, S., Low, G.: Comparison of bio-inspired algorithms for peer selection in services composition. In: Proceedings of the IEEE International Conference on Services Computing, pp. 250–257. IEEE Computer Society (2011)

[43] Shneidman, J., Parkes, D.C.: Rationality and self-interest in peer to peer networks. In: Proceedings of the 2nd Int. Workshop on Peer-to-Peer Systems (IPTPS03) (2003)

[44] Sigmund, Hauert, C., Nowak, M.: Reward and punishment. P Natl Acad Sci USA (19), 10,757–10,762 (2001)

[45] Sigmund, K.: Punish or perish? retaliation and collaboration among humans. Trends in Ecology and Evolution **22**(11), 593–600 (2007)

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

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

[48] Sun, Q., Garcia-Molina, H.: 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), pp. 506–515. IEEE Computer Society, Washington, DC, USA (2004)

[49] Villatoro, D., Sabater-Mir, J., Sen, S.: Social instruments for robust convention emergence. In: T. Walsh (ed.) Proceedings of the International Joint Conference on Artificial Intelligence, pp. 420–425 (2011)

[50] Zhang, H., Croft, W.B., Levine, B., Lesser, V.: A multi-agent approach for peer-to-peer based information retrieval system. In: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1, pp. 456–463. IEEE Computer Society (2004)

[51] Zhong, M.: Popularity-biased random walks for peer-to-peer search under the square-root principle. In: Proceedings of the 5th International Workshop on Peer-to-Peer Systems (IPTPS) (2006)