Mechanisms to promote cooperation in decentralized service*

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

Departamento de Sistemas Informticos y Computacin,
Universitat Politcnica de Valncia, Camino de Vera S/N, Valencia, Spain
{edelval, mrebollo, vbotti}@dsic.upv.es

Abstract. In distributed environments where entities only have a partial view of the system collaboration plays a key role. In the case of decentralized service discovery in 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 reject forwarding queries from other agents; therefore, the efficiency of the decentralized service discovery could be seriously damaged. In this paper we propose the combination of structural changes and incentives based on utility in order to promote cooperation in the service discovery process. The results show that, even in scenarios where the predominant behavior is to not cooperate cooperation emerges.

Keywords: Incentives, cooperation, service discovery, 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. 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 [5]. 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.

* An extended version of this work has been published in Journal of Systems and Software

One of the scenarios where cooperation plays an important role is service discovery in open Service-Oriented Multi–Agent systems (SOMAS) [1]. These systems are populated by agents that offer their functionality through services. 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. 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 [10]. 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 [9]. *Punishment* has also been considered to promote cooperation and to overcome the 'tragedy of the commons' [5]. 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 [6]. In general, punishment has been proven to be an efficient way to maintain cooperation [12].

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 [11, 7].

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. [3] present a model that uses the Prisoner's Dilemma game 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 in sustaining cooperation. Griffiths et al. [4] 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 [13]. 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.

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 combination of mechanisms that we propose is based on social plasticity and incentives. The obtained results show that, even in adverse situations where there is a large number of non–cooperative agents, our proposal obtains better results than other mechanisms proposed in the literature and the performance of the system is not seriously affected. 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.

2 System Model

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* [8, 2]. 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 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.

Agents are characterized by the set of services they provide and an internal state, where they store information about their degree of cooperation and the activity related to the discovery process.

Definition **2** *(Agent). An agent* $a_i \in A$ *is characterized by a tuple* (S_i, N_i, st_i) *where:*

- $S_i = \{s_1, \ldots, s_l\}$ *is the set of semantic service descriptions of the services provided by the agent;*
- 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,*

Network Creation Process 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*[8], which is considered to be self–organization criterion [2]. The homophily concept translated to the agent context has been considered as the similarity between two agents based on service information. Two agents in the system have a high degree of homophily if they offer similar services. The degree of homophily between two agents $H(a_i, a_j) \in [0, 1]$, where the higher the homophily value is, the more similar the agents are. 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. When an agent arrives to the system, it must be joined to an existing agent. The result of using this criterion to establish links between agents is a growing network structure based on homophily that has an exponential distribution of its degree of connection. This structure has certain small–world characteristics that facilitates the task of decentralized service discovery only considering local information. For a detailed mathematical treatment about how homophily is calculated and the network creation process, we refer the reader to [2].

Service discovery process The service discovery process relies on the cooperation of the agents. This process starts when an agent needs to locate an agent that offers a certain service in order to deal with one of its goals. Initially, agent a_i creates a query with the terms of the service.

The discovery process follows a hill climbing algorithm, in which the query is redirected exactly to one of the neighbors until the query arrives to an agent who provides a service similar enough to the required service. The selection of the most promising neighbor is a function that depends on the similarity of the neighbor and its degree:

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

This formula uses the same homophily–based factor (H) that has been used in the network formation [2] and a degree–based factor (number of neighbors $|N_j|$)

to select the most promising neighbor. The divisor of the expression is just a normalization factor. This process continues until a suitable agent is found or the number of times the query has been forwarded is over a maximum bound or *Time to Live* (TTL).

When self–interest is introduced in the system, agents decide to cooperate (to pass the service query to one of their neighbors) or not to do it. In the latter case, an alternative path must be located. As a result of this non-cooperative behavior, the number of steps required to reach the target agent a_v increases considerably. 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. These mechanisms should promote and maintain cooperation in the system in order to guarantee the proper performance of the system.

2.1 Structural Mechanism: Social Plasticity

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. 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* [3]. 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.

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 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 most influential constant is the displacement d, that 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

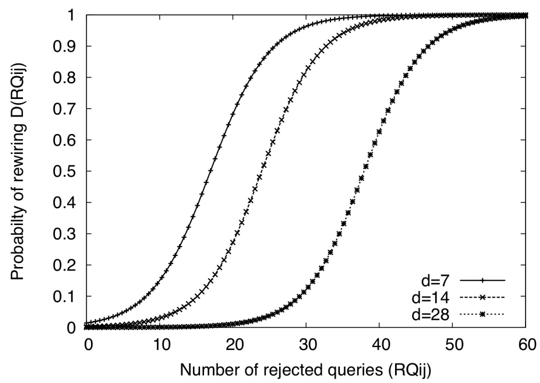


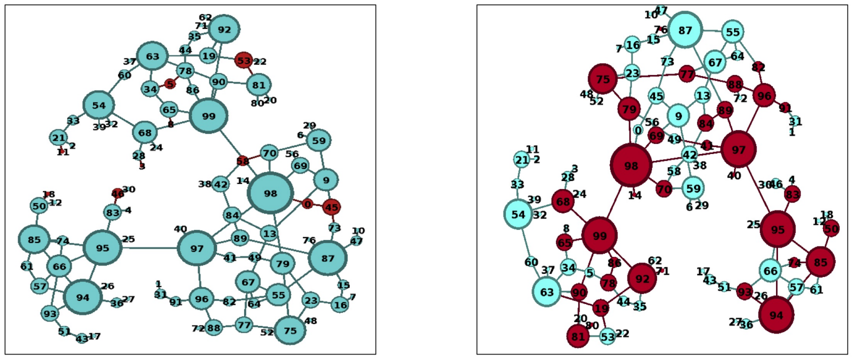
Fig. 1: Sigmoid function that calculates the probability of breaking a current link with a neighbor and looking for a new neighbor. The figure shows the shape of the function with different values of displacement parameter d.

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 1). y parameter indicates how abrupt the change of behavior is (the lower y value is, the more steepy the slope is). 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.

The agent a_i uses the D function and considering this probability it decides to break its current link and look for a neighbor 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. 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*;
– looking for a *similar neighbor to the previous neighbor*.

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 the system. However, the use of structural mechanisms when the number of non–collaborator 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



(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.

Fig. 2: 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.

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.

2.2 Incentive Mechanism

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 cooperators, 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:

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

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. 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.

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 assess 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 2, 75% of the agents were cooperative and 25% were non-cooperative. 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 2, non–cooperative agents are represented by red nodes and cooperative agents by blue nodes.

Figure 2a 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 2b 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.

2.3 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.

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. Based on this comparison, the agent decides 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 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.

	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)

Fig. 3: Payoff matrix: (Left) Prisioner's Dilemma, (Right) Stag Hunt Dilemma.

Finally, if a_i finds a neighbor that cooperates, the neighbor increases by one its local information about the number of queries forwarded. 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.

3 Results and Discussion

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. 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 was successfully solved when the degree of semantic match between the semantic service descriptions was over a threshold $\varepsilon = 0.75$ with $TTL = 100$. 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.

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 $y = 4$.

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

- **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)**: 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. This idea is similar to the Win or Lose Fast (WOLF) method, but the WPL method offers better performance than WOLF [14].
- **Game Theory.** We considered two type of games: the Prisoner's Dilemma (**PD**), where individuals might not cooperate even though it seems to be their best interest to do so; and Stag and Hunt (**SH**), which describes a conflict between safety and social cooperation. The main difference between them is the payoff matrix (see Figure 3). In these games, cooperate implies forwarding queries and not cooperate rejecting forward queries. 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 $p = 0.01$.

The tests 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.

When the number of collaborative agents is high enough, incentive–based methods has similar performance and 'SH', 'SP' and 'Incentives+SP' obtained similar results. The main advantage of the combined method 'Incentives+SP' is that the average path in the query resolution is sorter and the number of structural changes is lower, so it seems to be more efficient (see [2] for a detailed analysis). But the difference increases 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.

Figure 4 (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

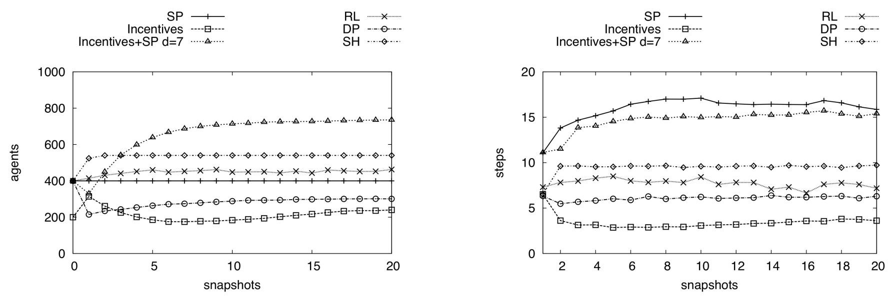


Fig. 4: Evaluation of (Left) the evolution of cooperation in the system, and (Right) the average path length in the discovery process when there are 400 cooperative agents and 600 non-cooperative agents.

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 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 4 (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 5 (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 5 (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 'SH' mechanism also improved the success rate, but the improvement was not as significant as the improvement achieved by 'Incentives+SP' or 'SP'.

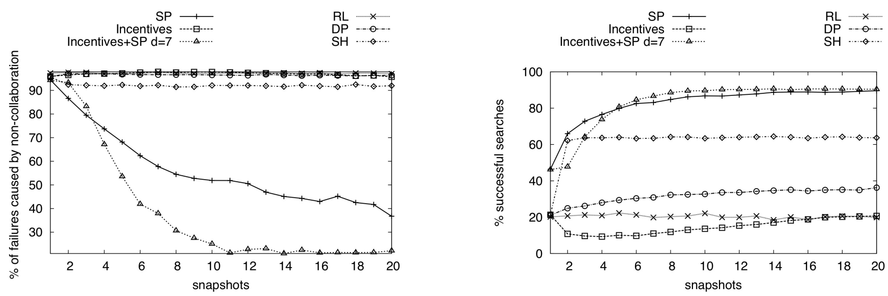


Fig. 5: Evaluation of (Left) the percentage of failures because of the absence of cooperation, and (Right) the percentage of searches that end successfully in the system when there are 400 cooperative agents and 600 non-cooperative agents.

4 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.

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.

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.

References

1. 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)
2. 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)
3. 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)
4. 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)
5. Hardin, G.: The tragedy of the commons. Science (162), 1243–1248 (1968)
6. 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)
7. 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)
8. Lazarsfeld, P.: Friendship as a social process: A substantive and methodological analysis. Freedom and Control in Modern Society (1954)
9. Nowak, M.A., Sigmund, K.: Evolution of indirect reciprocity by image scoring. Nature 393(6685), 573–577 (1998)
10. Nowak, M.A.: Five Rules for the Evolution of Cooperation. Science 314(5805), 1560–1563 (2006)
11. 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)
12. Sigmund, K.: Punish or perish? retaliation and collaboration among humans. Trends in Ecology and Evolution 22(11), 593–600 (2007)
13. Villatoro, D., Sabater-Mir, J., Sen, S.: Social instruments for robust convention emergence. In: Walsh, T. (ed.) Proceedings of the International Joint Conference on Artificial Intelligence. pp. 420–425 (2011)
14. 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)