Minds & Machines (2013) 23:419–441
DOI 10.1007/s11023-013-9304-8

Gossip-Based Self-Organising Agent Societies and the Impact of False Gossip

Sharmila Savarimuthu · Maryam Purvis · Martin Purvis · Bastin Tony Roy Savarimuthu

Received: 20 February 2012 / Accepted: 21 January 2013 / Published online: 3 February 2013

Abstract The objective of this work is to demonstrate how cooperative sharers and uncooperative free riders can be placed in different groups of an electronic society in a decentralised manner. We have simulated an agent-based open and decentralised P2P system which self-organises itself into different groups to avoid cooperative sharers being exploited by uncooperative free riders. This approach encourages sharers to move to better groups and restricts free riders into those groups of sharers without needing centralised control. Our approach is suitable for current P2P systems that are open and distributed. Gossip is used as a social mechanism for information sharing which facilitates the formation of groups. Using multi-agent based simulations we demonstrate how the adaptive behaviour of agents lead to self-organisation. We have tested with varying the gossip level and checked its impact in the system's behaviour. We have also investigated the impact of false gossip in this system where gossip is the medium for information sharing which leads to self-organisation.

Keywords Self-organising systems · Freeriding problem · Peer-to-Peer cooperation · File-sharing · Multi-agent systems · Artificial societies

Introduction

One of the most common problems in P2P (peer-to-peer) networks is free riding (Saroiu et al. 2002). In our context, free riders are those agents that do not contribute to the collective goals of the networked society, but make use of the

S. Savarimuthu (✉) · M. Purvis · M. Purvis · B. T. R. Savarimuthu
Department of Information Science, University of Otago, Dunedin, New Zealand
e-mail: sharmilas@infoscience.otago.ac.nz

resources of the network. These free riders decrease the overall performance of the society by degrading the common good (Hardin 1968).

Electronic societies suffer from these free riders who exploit the common resources (e.g. bandwidth in a file sharing system). Many existing approaches employ centralised social regulations to control free riders. Researchers have used monitoring agents or governor agents to control agent behaviour (Purvis et al. 2006). But these centralised mechanisms are computationally expensive for a system. Centralised mechanisms are known to cause performance bottlenecks and also suffer from scalability issues (Saroiu et al. 2002).

The openness of the Internet allows users to dynamically join and leave the system at any point of time (Kempe et al. 2003; Jelasity et al. 2004, 2005; Gorodetsky et al. 2007; Zaharia and Keshav 2008; Dasgupta 2003; Boyd et al. 2006). P2P systems are increasingly attractive due to continually improving bandwidth and processing capabilities of distributed and mobile platforms. The clients of file sharing systems are not only personal computers but also smaller devices such as smart phones. There is a need for decentralised solutions to deal with the free riders. The solution to the free-riding problem should take into account the open, dynamic and distributed nature of modern software systems.

This paper proposes a decentralised solution that makes use of social mechanisms such as gossip (Rebecca 1956) and ostracism (Thomsen 1972). The inspiration to use social mechanisms for our work comes from the human societies, which have evolved over millennia to work effectively in groups. For human beings group mechanisms provide social machinery that supports cooperation and collaboration. Social control is a fundamental concept that has evolved in human societies. Social control can be employed through leadership mechanisms. For example, the leader can impose rules on his followers. The disadvantage of this approach is that it is centralised. On the other hand, social control can be achieved using a bottom up approach.

For example, a gossip-based mechanism can be used to achieve social control as it serves as a distributed referral mechanism where information about a person or a group is spread informally among the agents. This approach can be used to achieve control in agent groups. Another social mechanism that can be employed to deal with free riders is ostracism. Members that do not adhere to the values or expectations of the groups can be sanctioned by the other agents by their refusal to interact with those agents.

In this work we demonstrate how these social mechanisms can be employed in an open, dynamic and decentralised society where several groups are formed and are ranked based on their performance.

The aim of the work discussed in this paper was to develop a self-organising open and dynamic system, where new agents may come into the society and also agents may leave the society at any time. A truly open and dynamic system will allow the formation of new groups and dismantling of existing groups according to the population size. Our aim was to achieve that in a decentralised manner without explicit control at the top level. Forming groups using tags is helpful, since it is scalable and robust (Hales 2004). For higher numbers of peers, more tag groups can be formed, and that process will scale well for any number of peers (i.e. if there are more peers, more groups will be formed, and if there are fewer peers, fewer groups will be formed).

The remainder of the paper is organised as follows. The social concepts used in this work are introduced in Sect. "Experimental model". Our experimental setting and selected experimental results are described in Sects. "Experimental setup" and "Results". In Sect. "Related work and comparison" we present the related work and the comparison with our previous work. Finally, Sect. "Future work" concludes the paper.

Experimental Model

Our experimental model presents a social situation in which the agents have the option to share or not to share (2-person dilemma) in a pairwise interaction within their group.It is a 2-person dilemma. Sharing would cost the donor who shares. But the receiver receives the benefit without incurring any cost. Non-sharing (defection) is the selfish option which benefits the individual but is not good for the society. Sharing benefits the society by improving the performance of the whole system, which leads to the overall betterment of the society. Since the donating agent spends some effort (e.g. bandwidth) in the process of donating, it incurs some cost in our model. That sharing agent could have decided to be selfish and thereby avoid incurring that cost. Thus free riding becomes a threat to the society, causing damage to the common good. This is the issue of the "Tragedy of the Commons" (Hardin 1968). A brief overview of the social mechanisms used in our experiments to deal with free riding are described below.

Gossip

Gossip is a powerful mechanism in human society for information sharing. Research done by evolutionary biologists suggests that humans have shown more interest in gossip more than in the original information (Sommerfeld et al. 2007), when participants were presented with both types of information (the gossip information and the "real", original information). Based on their research they have noted "gossip has a strong influence... even when participants have access to the original information as well as gossip about the same information" and also that "gossip has a strong manipulative potential".

In a way, a gossip mechanism can be considered to be a 'distributed referral' mechanism (Eugster et al. 2007). It is similar to having a reputation system, except that the gossip information is distributed. Paine (1967) has explained that people who gossip within their gossip circle feel the fellowship or belongingness to the community. In fact gossip is a property of a group (Paine 1967) that can be used to provide local social control; it aids in maintaining the social structure. Gossip can also be considered as an indirect attack on a person where no other way of sanctioning is easily possible. Thus gossip is a mechanism for transmitting public opinion which can lead to some social benefit (Rebecca 1956). Gossiping is also a way of doing social comparison (Suls 1977; Paolucci et al. 2007).

The grooming behaviour in animals is analogous to the gossip behaviour in humans (Dunbar 1996). Dunbar's work (Dunbar 2004) also acknowledges that gossip is a way of social bonding, and it is a part of social life. An additional benefit of gossip is that it helps to control freeriders. Gossip information about freeriders is important, because people do not want to be exploited by a freerider, and freeriders degrade the societal welfare. The most common way of controlling freeriding is based on memory of past behaviour (Dunbar 2004). Gossip helps this by being a medium for 'information storage and retrieval' (Roberts 1964).

The work of Conte and Paolucci (2002) considers gossip as a medium to spread and retrieve the reputation about an agent which aids in establishing social control.

de Pinninck et al.'s (2008) work has adopted the gossip mechanism to achieve a similar goal as ours. In their work gossip has been used as an identification mechanism to spread and find information about the norm violators. In our work we use gossip for a similar purpose, which is to identify freeriders. In their work (de Pinninck et al. 2008), gossip is shared locally with nearby agents called 'mediators', and it does not need complex computations. The mediator agent contacts the enforcer agent to punish the violator. Gossip has been used in relation to the norm enforcement technique in their work, and gossip has been also used for reputation management in Yu and Singh's (2000) work.

In a recent work of Mordacchini et al. (2010), they have proposed an architecture for gossip-based peer-to-peer systems which facilitates information exchange among peers through gossip. Their system groups users with similar interest (or users who are interested in similar contents) by the distributed process of sharing information through gossip. This sort of grouping has an advantage of making recommendations accordingly, since those groups of users share a common interest. It is different from our work, since our work groups agents based on behaviour (cooperativeness). The work of Khan and Tokarchuk (2009) proposed a group-structured P2P system which also uses gossip to form groups with peers of common interest. The work of (Khan and Tokarchuk 2009) has used gossip to form groups based on peers looking for similar content (or peers with common interest). In our work we too use gossip to form groups (based on behaviour), but content searching is not the focus in our work.

The work by Ganesh et al. (2003) has proposed a similar gossip-based information dissemination for P2P membership. It has used a gossip-based distributed mechanism which does not require a global view, so it uses less memory and leverages scalability and decentralisation. These aspects are similar to the approach used in this work, but our focus is not P2P membership. Instead our focus is on forming dynamic groups automatically by enabling self-organisation of agents into groups of different levels of cooperativeness to avoid exploitation by freeriders.

In Nowak's model (Nowak and Sigmund 1998) of donor-receiver game, the donor chooses the strategy based on the past reputation (image score) of the receiver. Players use the experience of others (to know about the receiver). As individuals tend to cooperate with the good ones and defect the bad ones, this model establishes cooperation based on indirect reciprocity. Our model also has this notion of sharing experience with others through gossip mechanism, since the reputation (image score) of the player is not directly visible/accessible.Our model aims for group segregation by separating good and bad in order to restrict exploitation in electronic societies.

There is other work in agent-based simulation and P2P systems which has used gossip-based protocols (Kempe et al. 2003; Jelasity et al. 2004, 2005; Gorodetsky et al. 2007; Zaharia and Keshav 2008; Dasgupta 2003; Boyd et al. 2006). Gossip-based protocols broadcast information, and they are based on the way information spreads in the form of rumors in human societies. The information is shared between one-to-many individuals. In these protocols the number of participants (with whom the information is shared) increases each time. In this fashion the information is spread to everyone in the society in a short period of time. The disadvantage is that lots of resources are used for this sort of information sharing without realising whether the information would be useful to the recipient. This results in information flooding which overloads the system (i.e. causes congestion and performance bottlenecks). Instead, it could be made more simple and useful if the recipient could get the information by requesting, rather than having all types of information being overloaded through a flooding model. Following this request-based approach, the user has an option of choosing the information he/she wants and avoiding the unnecessary information. This approach does not overload the system, and it needs less computation compared to the former approach. Moreover, the user has the control over the information he/she is being exposed to. Our approach uses such a kind of gossip mechanism where the user can specifically request for information that he/she needs.

Ostracism

It has long been the case in human and animal societies that the member of a group who does not abide by rules or norms can be punished by other members of the groups (the followers of the rule/norm). One kind of punishment is ostracism (Thomsen 1972), which results in the social exclusion of the punished member. All the other members of the society would stop interacting with the member who is being ostracised and would no longer consider that person to be a part of their group (by ignoring him/her). This kind of behaviour is used as a social punishment mechanism where there is no higher authority or institutional monitor to check deviations and establish punishment. Thus it is a decentralised mechanism as opposed to having a central controlling authority to establish sanctions.

There is work which has used ostracism for improving cooperation where the context of interaction has been modeled using the Iterated Prisoner's Dilemma (IPD) game . In Hirshleifer's work (Hirshleifer and Rasmusen 1989) cooperation is achieved by a grouping mechanism. Players who persistently defect are expelled from the group as a way of punishment by other players. Thus group ostracism is used to protect cooperators from the defectors. By imparting the fear of punishment, cooperation is enforced. Thus having ostracism as a punishment mechanism has helped to control defectors and promote cooperative behaviour in the IPD game. Oh's (1999) work also used ostracism along with a grouping mechanism in the context of modeling interaction using an IPD game. Ostracism is used to expel defectors from the cooperative groups. By doing this, cooperators would only have other cooperators in their groups and they play only with them. Thus cooperative groups of similar players are formed and emergence of cooperation is achieved.

Pinninck's work (de Pinninck' et al. 2008) has used an ostracism-based mechanism to punish norm violators in an open multi-agent system. Their work makes use of the concept of a normative reputation for each of the agents in the society. Depending upon whether an agent abides by the norm, its reputation is spread through gossip. Agents with a bad normative reputation are ostracised by the members of the society (i.e. agents stop interacting with a norm violator). By ostracising the norm violators, agents achieve better payoffs by interacting only with the normative agents. Thus the norm is enforced, and the norm violators are punished in an open agent society. Similarly, our work also uses ostracism along with gossip to identify freeriders and restrict exploitation in a simulated P2P environment.

Experimental Setup

In our experimental arrangement agents are engaged in the sharing of digital goods in a P2P environment of a simulated artificial agent society. The system is developed as a distributed system without central control.

Agent Attributes

For this experimental model we have used the agents which have fixed, randomly assigned attribute values which represent how they behave.

- *Cooperativeness value* This attribute concerns how cooperative an agent is. Agents have a randomly assigned cooperation value between 0 and 10 that represents how much they cooperate (share), with 0 representing an agent that never cooperates and 10 representing an agent that cooperates every time. This value is known as the cooperativeness of the agent.
- *Tolerance value* Agents have a tolerance value between 1 and 10, which characterizes how much non-cooperation the agent can tolerate before it decides to leave the group. A value of 1 identifies the least tolerant agent, and 10 identifies the most tolerant agent.
- *Rejection limit* Rejection limit represents how many rejections the agent can face before it decides to leave for another group.
- *Gossip blackboard length* Each agent has a gossip blackboard of certain length to store the gossip messages from other agents of its group. Each agent also has a memory of certain number of previous groups to which it belonged.
- *Life span* Agents are set to have life spans, which determine how long the agents remain in the society. When an agent's life span is over it leaves the society (i.e. die).
- *Cost and benefit for sharing* A sharing agent loses 0.1 as cost for sharing and the receiving agent receives 1 as benefit.

Table 1 Experimental parameters for open society

Experimental parameters	Values
Number of agents to start with	100
Number of groups to start with	5
Number of iterations	500
Agent's cooperative value	0–10 (random)
Agent's tolerance value	1–10 (random)
Agent's rejection limit	10
Agent's gossip blackboard length	10
Agent's group memory limit	5
Agent's lifespan	Varies
Number of gossip requests	5
Group's size for dismantling	5
Group's size for splitting	40
Cost for sharing	−0.1
Benefit for receiving	1

Experimental Parameters

In the initial setup agents are put into random groups. Each group can be imagined to be represented by a tag (badge). Agents within a group have the same tag. They interact within their group, and they can also move to other groups under certain conditions. In such cases they join the other, jumped-to group, and the tag changes accordingly. Agents can ask for gossip feedback about other agents' behavior. Groups are formed or dismantled based on their size. The procedure of the experiment is explained in the following sections. The experimental parameters are listed in Table 1.

The procedure of the experiment is explained below.

Publishing Gossip

In each iteration, a certain number of random players (agents) may ask for files from other players of their group. A player can gossip about the outcome of an interaction with another agent (random) in its group (report whether the other agent was cooperative or not). In this gossip mechanism we assume that there is no lying. Since this happens within the group (agents in a group have same tags), we have assumed that the agent has no motivation to lie. In this fashion, every transaction is reported (gossiped about) to one of the other agents in the group. Thus the overall system has some partial information about the cooperativeness of each agent, maintained in a distributed way. For further illustration, the operation of how peers publish gossip is explained in the following example. Consider A, B and C as the three random agents in a group. A is the taking-player, B is the giving-player and C is the gossip holder. A asks for a file from B. If B shares then A gossips positively about him to C, otherwise A gossips negatively about him to C.

Using Gossip

Each peer has a limited amount of memory space for storing new gossip information. After reaching the storage limit, the memory register rolls over, based on a First-In-First-Out (FIFO) algorithm. When a player requests a file, the giving-player can check with a certain number of (e.g. five) other random agents (asking them what they know from the gossip information they have received) whether this taking-player is the worst cooperator of their group. The worst player is the one who has been uncooperative most times in its group (according to the available gossip information). If the taking-player is the worst player, the giving-player refuses to interact with the taking-player (ostracism). Otherwise this giving-player interacts (sharing a file or not based on its own cooperativeness). The operation of how peers use gossip is explained by the example given below. Assume X and Y are the players in the group where X is the taking-player, Y is the giving-player. Y checks with five other players in the group in order to see whether X is the worst player in their group. If so Y refuses to play with (share file with) X. Thus X is ostracized. Otherwise Y plays with X. When only a few agents (less than five) have gossip about the taking-player, then only the available information is taken into consideration. Sometimes it can be the case that none of the players have gossip about the taking-player. In such a case the taking-player is not considered to be the worst player, a privilege similar to what happens when a new player joins a group. By this process agents share their file, taking gossip into consideration which is about other agents' past behaviour.

Leaving a Group

An agent can leave a group for two reasons. A player can leave a group if its tolerance level is surpassed or its rejection level is surpassed. We call this leaving agent a "hopping peer". If its tolerance limit is reached, that means this agent is in a group where others do not cooperate at the rate that meets this agent's minimum level of expectation. Thus after a number of such non-sharing events from the group members (the agent's tolerance limit is surpassed) the agent will decide to leave that group and move to another group. If its rejection limit is reached, that means this agent is in a group where it is considered to be the worst cooperator by some other agents so it has been refused a play more often than others. If the rejection level is met then the agent will leave that group and move to another group.

Choosing a New Group to Join

When an agent decides to leave a group and join another, it looks for a group that may accept it. Agents can apply to enter into other groups they choose but they get entry into a group which matches its cooperativeness. A good agent would get into a group that is better than its current group while a bad agent should get into a group that is worse than its current group. This process is explained in detail in (Savarimuthu et al. 2010). We have restated it in the following paragraphs.

Table 2 Previous group history

Group no	Iteration no	Cooperativeness
1	560	4.5
3	700	6.0
2	1,200	6.4

The hopping peer collects information about other groups from their group members. Then it decides to which group to request admission from. Every agent has a memory record of its most recent groups (in our experiments the memory limit was set to 4). For example, assume agent E has been in 3 other groups before, as shown below in Table 2.

The first row of Table 2 shows that E has left group 1 at the 560th iteration, and the cooperation value of that group was 4.5 at that time. E left group 3 at the 700th iteration and group 2 at 1,200th iteration. Since the composition of groups change over time, the cooperativeness of the group also changes. So it is likely that the most recent information will be the most accurate and useful for an agent. Since all agents have a memory of their previous groups, the hopping peer can collect this information from all its group members and calculates the latest information about other groups. In particular, the agents who moved into this group recently from other groups have the most recent information. Taking this information into consideration, the agent decides where to move. For example assuming the current iteration is 1,400, the latest information collected from the group members is given in Table 3.

Assume here that agent L intends leaving group 4, and Group 4's cooperativeness is 6.6 at that moment. From the latest information agent L knows about other groups and their cooperation value. For agent L, groups 5 and 3 are better, since the cooperation value in those groups appear to be higher than L's current group. Groups 2 and 1 are lower-ranked groups. So agent L chooses to move to the groups in the order of their ranking.

If L is intolerant of its current group (which means it is not happy about the cooperativeness of its current group), it will try to enter into the best group that it can find. This is the case of an agent being "too good" for its current group and wanting to move to a more cooperative group. But if the better groups on its list does not allow entry, then the intolerant agent L may determine that there is no group available that is better than its current group, and it will remain in its current group. In this case its tolerance limit is reset to 0.

On the other hand, an agent may not be good enough for its current group (i.e. it is being shunned by the other members for being the worst member of its group).

Table 3 Latest available information

Group no	Iteration no	Cooperativeness
5	1,330	8.1
3	1,170	7.5
2	1,200	6.4
1	1,199	3.8

Because of refusals from other agents to play, its wealth will not increase, and it will want to leave and find some another group in which it can find players to play with. If the better groups do not allow entry, the agent will go to lower groups, since it is better off moving to any new group (even if it is a lower group) rather than staying in the current group where it is known as the worst player. How a player gets entry to another group is explained in the following section.

Joining Another Group

The hopping peer asks any randomly chosen agent in the group to which it seeks entry for its permission to enter. We call this permission-granting agent in the group to which entry is sought, the "checking peer". The checking peer will accept an agent whose cooperativeness value is greater than or equal to a value calculated by a formula (given below). This formula is the same one used in our previous work (Savarimuthu et al. 2010). This hopping peer will gain permission to enter the group whenever its cooperativeness is greater or equal to the group's entry value calculated by the following formula.

$$\mathtt{EV} = \mathtt{C} - (\mathtt{C1}/(\mathtt{SL} - \mathtt{S})^{C2}) + \mathtt{C3}^{(S-SU)} \tag{1}$$

The group Entry Value (*EV*) is calculated considering the given group's calculated Cooperativeness (*C*) and its group Size (*S*). *C* is the group's calculated Cooperativeness through the gossip information, and *S* is the size of the group. *C1*, *C2*, *C3* are constants whose values in our experiments were 25, 2, 10, respectively. These constants were adjusted to make the *EV* expression appropriate for two "boundary values", the upper size limit of a group (*SU*) and the lower size limit of a group (*SL*). Accordingly, *C1* takes the value of *SU* and *C3* takes the value of *SL*. In our experiments, *SU* was set to be 25, and *SL* was set to be 10. It is inappropriate or inefficient for groups of players or traders to become too big or too small. If a group becomes too big, then it becomes unmanageable. Assume there is only limited resource or space available (Nowak and Highfield 2012, p. 88). If a group becomes too small, below a certain value, then the group is not considered to be an active group. There should be at least a certain number (minimum value) of players in it to be considered as a team (e.g. a sports team may have a certain number of players as a minimum for the existence of a team/group. Thus a volleyball team must have at least 6 players. If there are less than 6 players, then it is not considered to be a team/group). If the size of the group is 10 or below, the entry qualification value will be set to a low value, making entry into the group very easy to obtain. If the size is 25 or above the entry qualification value is set to a high value and that would make it difficult for any but the most cooperative agents to join. Any values of the *EV* expression that fall below 0 are set to 0, and entry values above 10 are set to 10. Thus a group's entry value is always between 0 and 10.

A simple example illustrates the use of this formula. Consider that a group's calculated cooperativeness (*C*) is 6. When the group Size (*S*) is 14 the group Entry Value (*EV*) is 4.43. When the group Size (*S*) is 25 the Group Entry Value (EV) is 6.88. In our system, the checking peer needs to get an estimate of the cooperativeness of the hopping peer (the agent seeking entry). So the checking peer asks 5 randomly chosen players from the hopping peer's group about the hopping peer's cooperation. It is thus inquiring into gossip information from the hopping peer's group. Consider a case where E and F are in different groups. E is the checking peer, and F is the hopping peer that wants to enter E's group. F asks E for entry, and E asks 5 other randomly chosen players in F's group for gossip information about F's cooperativeness. The averaged value is calculated (out of 10) from this information considering the worst case scenario. This estimated cooperativeness would be a value between 0 and 10. If F's estimated cooperativeness calculated through this gossip information is greater than or equal to the entry value (*EV*) of its group, the checking peer allows entry for the hopping player; otherwise it denies entry. In that case the hopping peer will try to enter into other groups. The hopping peer will ultimately get into a group where its cooperativeness meets the eligibility criteria to enter. If no such group is available, the hopping peer stays in its current group.

The entire process is repeated for many iterations, and gradually, some groups will emerge as elite groups with many cooperators, and other groups will have less cooperative players. As a consequence, these mechanisms achieve a separation of groups based on performance.

Groups Splitting and Dismantling

Our aim has been to develop a self-organising open and dynamic system, where new agents may come into the society and also agents may leave the society at any time. To start with, new peers are allowed to join the society by gaining entry into random groups in the society. They can build their way up to higher groups based on their cooperativeness. A truly open and dynamic system will allow the formation of new groups and dismantling of existing groups according to the population size. Our aim was to achieve the same in a decentralised manner without explicit control at the top level.

The agents' lifespan determines how long the agents remain in the society and when they leave (i.e. "die"). At any time a new agent could join the society and an existing agent could leave when its lifespan is over.

Since the number of agents in the society at any time is dynamic the system adapts itself to form new groups if more agents join. It also dismantles groups if there are fewer agents in the society (less than the lower size limit of a group).

The motivation for splitting and dismantling comes from real life societies. For example, when the size of a group becomes too large, it becomes unmanageable. Larger hunter-gatherer groups split because of reasons such as seasonal change or inequality in resource sharing (e.g. when meat is not shared equally).

In our approach, a group splits into two if the size of group reaches a certain limit (40). Based on the local gossip information in the splitting group, the top cooperators (first half) form one group and the rest (second half) form the other group.

If the size of the group decreases and goes below a certain limit (5) then the group dismantles. The remaining agents in the group go to random groups where they could enter. This is similar to a society where it can be functional only if the society has a certain size. For example, in hunter-gatherer societies, in order to hunt larger preys a group has to have a minimum size. Otherwise, the prey cannot be hunted.

It should be noted that the splitting and dismantling functionalities account for the scalability of the system and its robustness.

Results

Before we present the experiments we have conducted and the results obtained, we would like turn the attention of the reader to the work reported in (Savarimuthu et al. 2010) where the results of the closed society are presented. In this work, there were 5 groups in the beginning with 100 agents in the society at any point of time (closed society). Our current work shows how the agents in an open society, self-organise themselves into groups based on their cooperativeness. The parameters in Table 1 were used for all experiments, except the arrival rate and departure rate of the agents. These were varied in each experiment to show the dynamic behavour of the system (as noted in each experiment's section).

Experiment 1: Self-organisation in an Open Society

We have conducted experiments on an open system by varying the arrival and departure rate of the agents. For all the experiments presented in this paper we start with 100 agents in 5 groups initially. After that agents can join or leave the society at random. For this example, we started the experiment with 100 agents in 5 groups at the beginning and added more number of agents on the go. The arrival rate of agents was upto 2 agents from iterations 50 to 100 then upto 4 agents from iterations 300 to 350 then upto 2 agents from iterations 350 to 400. The agents leave (die) when their life span is over.

Figure 1 shows two graphs which share the same x-axis. The x-axis shows the number of iterations. In the top graph y-axis shows the cooperativeness of groups. Each diamond shown in the graph represents the cooperativeness of a particular group. For a given iteration number in the x-axis, the y-axis shows the cooperativeness of all the groups that were present in that iteration. For example, in iteration 100, there were 6 groups (represented by diamonds), with different levels of cooperativeness. The graph given in the bottom of Fig. 1 shows the total number of agents (alive agents) in the society for a given iteration. For example, in iteration 100 there were 130 agents in the society.

These two graphs together show the dynamic behavior of the system (the formation of new groups and dismantling of old groups). It can be observed that, at the start the groups had an average cooperativeness value of 5. As the number of agents increased, new groups were formed (iteration 100). As the number of agents decreased (iteration 200), the number of groups decreased. The separation between the good groups and the bad groups is distinct. When the total number of agents was about 40 in iteration 300, there were fewer groups. Note that the cooperativeness of these groups was about 5 at that point. As the number of agents in the societies then

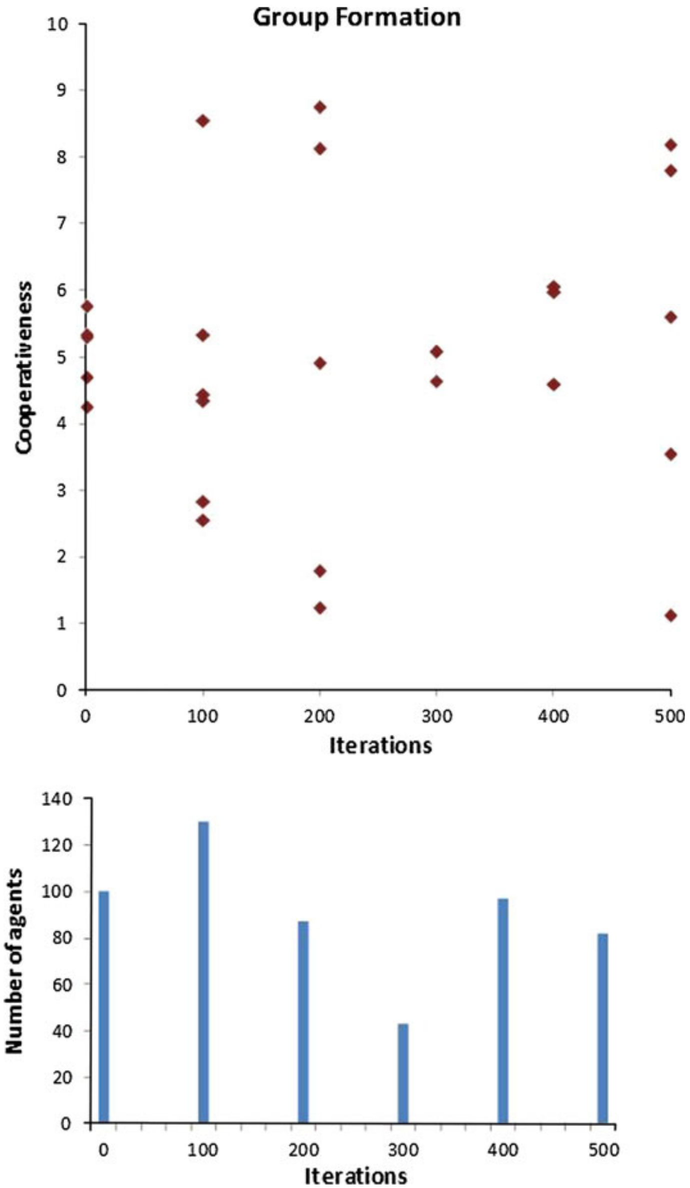

Fig. 1 Self-organisation of an open system when agents' arrival and departure rates are dynamic

increased, there were more groups and the separation between the good and the bad groups is evident. We note this process can be appreciated better by viewing the video (Savarimuthu 2010d).

There are two kinds of behavior we observe in the system. Firstly, the system dynamically enlarges or shrinks by creating more groups or dismantling existing groups based on the number of agents in the system. Secondly, it also forms groups based on cooperativeness. Cooperators move towards other cooperators and non-cooperators end up with other non-cooperators. The agents self-organise into groups that have different ranges of cooperativeness. Thus this system restricts the non-cooperators taking advantage of cooperators by restricting their access to better groups.

Experiment 2: Arrival Rate Greater than Departure Rate

We conducted experiments by keeping the arrival rate greater than the departure rate. A run of this experiment is shown in Fig. 2. Each group is represented by a square. It can be observed that when the number of agents increase, the system is able to create more groups dynamically and also these groups are separated based on the cooperativeness of the agents. This shows the scalability and self-organisning ability of the system.

Experiment 3: Arrival Rate Equal to Departure Rate

When the number of new comers is roughly the same as the number of leaving agents in the system, the system will have same number of agents and the number of groups remain the same. But new agents who join the society have certain cooperativeness. Because of this the composition of groups and the cooperativeness of groups change over time. Figure 3 shows the cooperativeness of five different groups over 500 iterations. The cooperativeness of these groups varies depending upon the net effect of the cooperativeness of the agents that are present in the

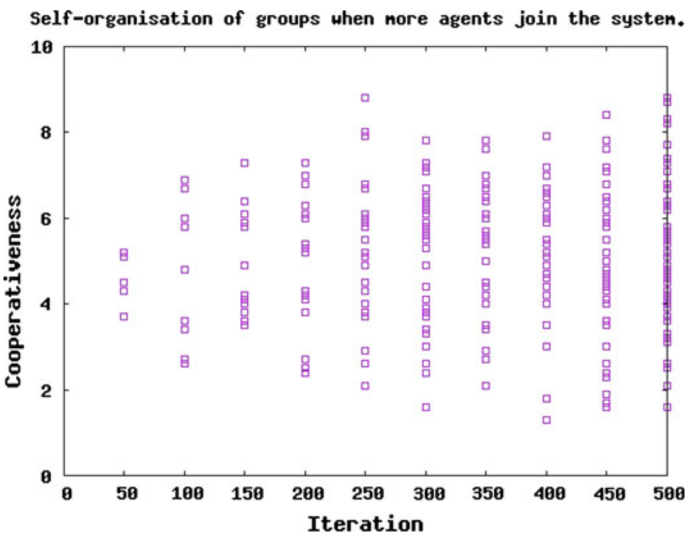

Fig. 2 Self-organisation of an open system when agents' arrival rate is increased

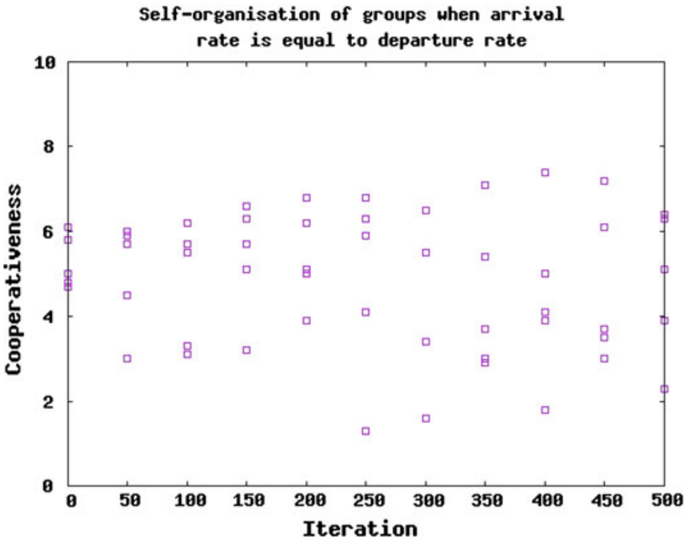

Fig. 3 Self-organisation of an open system when the arrival rate is equal to the departure rate of the agents

society. A new agent whose cooperativeness value of nine joining a group whose average cooperativeness value is five will increase the group's average. In the same way, a bad agent leaving a good group will increase the group's cooperativeness average. Figure 3 shows how the 5 groups change over time based on the number of agents (composition of the group) and the cooperativeness of agents present in the system over time.

Experiment 4: Varying Gossip Level

We also experimented by varying the number of agents that are contacted to collect gossip. This experiment was conducted with 100 agents. The number of agents from which gossip was provided varied between 2, 5 and 15. From the 30 sample runs collected, the means of the standard deviations of groups at the end of the experiments, for gossip sizes 2, 5 and 15 were 2.80, 3.22 and 3.23 respectively. We noticed significant differences when we compared gossip sizes 2 with 5. Collecting gossip from 5 agents resulted in better separation than from 2 agents. But when we compared 5 with 15 there was not much difference in the separation. Collecting gossip from 15 agents (or less, if there are less than 15 agents in that particular group) has slightly improved the separation, but the difference was very small. This suggests that partial information is sufficient to establish group separation, and thus one can avoid additional computation, because asking 15 agents instead of 5 will require much more message passing and computing in an agent society.

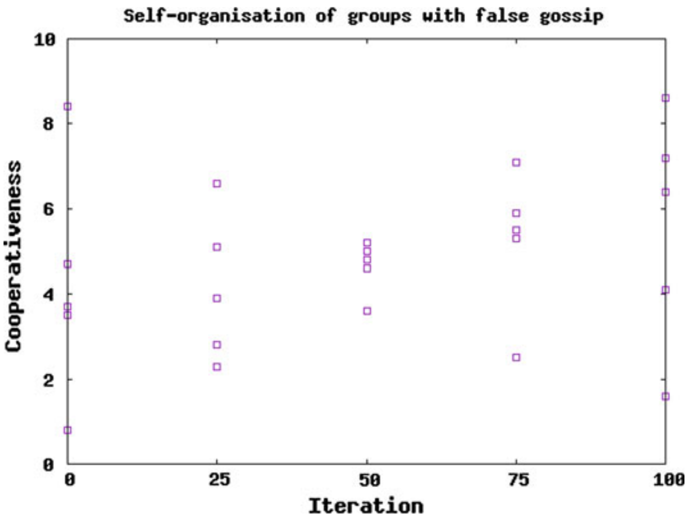

Fig. 4 The impact of false gossip

Experiment 5: Impact of False Gossip

We tried to experiment with false gossip to study its impact in the system's behaviour. False gossip is the opposite of the original gossip information (which means a lie). We experimented with the percentage of agents who give false information. This experiment was conducted with 100 agents. We varied the level of false gossip to 25, 50, 75 and 100 %. The result is shown in Fig. 4.

When there is no lying in the gossip the system self-organises and forms different groups with different cooperativeness. When there is 25 % of lying in gossip (or liars in other words) the system did not form well established groups with different cooperativeness like the no gossip system. But there is separation which is not as prominent as the no gossip system (0 %). When there is 50 % of lying in gossip the system was at its worst. There is no separation. When there is 75 % of lying in gossip, the system did not form well established groups with different cooperativeness like the no gossip system. It was similar to the 25 % of lying. When we tried 100 % of lying in gossip (100 % means eveyone is a liar in the system), the system self-organises again based on the false gossip which showed different groups with different cooperativeness.

It is because when the system has 100% false gossip the agents were understood to have completely the opposite of what their cooperativeness is. For example, an agent with 7 cooperativeness will be known as the one with 3 (10–7) cooperativeness. In a similar way he will end up in a group with other agents like him (7 known as 3, 8 known as 2 and so on). This forms groups of agents with similar cooperativeness even though the system has 100% false gossip.

Related Work and Comparison

In our previous work (Purvis et al. 2006), the self-organisation of peers in different groups was achieved by making use of tags and monitoring agents, where the population had a mixture of cooperators and non-cooperators. By employing a monitoring agent for each group, the system evolved into groups partitioned according to the performances of their group members. Each monitoring agent employed a voting mechanism within the group to determine which agents were the most and least cooperative members of the group. Then the most cooperative member was allowed to move to a new group, and the least cooperative member was expelled from the group. Those peers who left voluntarily or those who were expelled from their groups obtained membership in a new group only if the local monitor agent of the other (new) group accepted them. Since the local monitor agents picked players for their group based on performance, the high performing player had a good chance to get entry into the best group, and the reverse conditions applied for the worst performing player. As a result, the players entered into groups based on their performances. Though this system produced good results, this approach is semi-centralised, because it required a local monitoring agent for each group. In addition the work considered a closed society. We believe this system can be applied in a regimented, closed society but cannot be applied to the modern systems which are open and distributed.

Hales's (2004) work, extends his previous work on tags to networks, considers a 'neighbor list of nodes' as a tag. The 'movement of node in a network' is modeled as a mutation. His results showed that tags work well for P2P systems in achieving cooperation, scalability and robustness.

Pacheco et al. (2006) have shown that helping good individuals and not helping bad individuals increases the reputation of individuals and also refusing to help good individuals and helping bad individuals lead to a bad reputation in a group setting which they call as stern judging which is another model for indirect reciprocity. However this work does not consider the impact of false gossips (i.e. spread of false information) and also does not investigate the role of agents joining and leaving a society.

The work of Ohtsuki and Iwasa (2004) identifies conditions under which individuals in groups maintain cooperation under indirect reciprocity. There were eight such conditions and all these eight conditions exhibited two properties. (1) Cooperation with good people is regarded as good and defecting good people is considered bad and (2) defecting bad people is considered as good. Pacheco et al. (2006) have shown that one of the eight conditions identified by Ohtsuki and Iwasa, the 'stern-judging' condition (helping good individuals and not helping bad individuals increases the reputation of individuals and also refusing to help good individuals and helping bad individuals lead to a bad reputation in a group setting) is favoured over all the other conditions. The limitations of these two works are that they do not consider the impact of false gossips (i.e. spreading false information) and also does not investigate the role of agents joining and leaving a society. The work reported in this paper addresses these two issues. Also, Nowak (2006) has proposed five different ways of solving the problem of cooperation. However, the two above mentioned issues addressed by our paper has not been explicitly considered by his work.

Nakamaru and Kawata (2004) have shown that when a member only spreads gossips received from other players who have cooperated with him in the past, then this strategy cannot be invaded by lying agents. The group composition in their set up is fixed. However, in our set-up the group composition changes over time (i.e. agents come and go). In the future, we intend to employ their mechanism in our dynamic setup and study the implications. Smead (2010) used a simplified model of used in the work of Nakamaru and Kawata to investigate the coevolution of signalling (i.e. punishment) and indirect reciprocity. Note that Smead's work also does not consider a dynamic set-up of agents. Moreover, the investigation of the emergence of punishment signal is an interesting problem, but it is outside the scope of this work.

Skyrms and Pemantle (2000) and Skyrms (2009) investigate the dynamic evolution of social networks to maintain cooperation. Our work differs from these works in several ways. First, agents in their model do not make use of gossip information, but they make use of their own experience with other players (i.e. depending upon whether they were treated nicely in the visit they made to other agent's place). The agents cut off their link to other players if the interaction was not pleasant. In our work an agent ostracizes another agent based on the gossip information. Second, false information is not considered by these works while our work investigates the impact of false gossips. Third, our work considers dynamic evolution not in terms of network topology but in the form of agents joining and leaving groups and the splitting and dismantling of groups. In the future different types of network topologies can be considered for agent to agent interaction within a group.

In our present work, instead of the Prisoner's Dilemma game, we have adopted the more practical scenario of sharing digital goods in electronic societies. We investigate how a society can achieve the separation or self-organisation of groups in a decentralised manner in an open society. Such a system would help to protect cooperators from being exploited by the non-cooperators. It would also restrict the non-cooperators from taking advantage of cooperators by restricting their entry to better groups where the access to resources is better. Hence, the quality of service (e.g. the quality of file sharing) and the performance (e.g. utility of agents) in the better groups will be higher. By doing so, the performance of the whole system can be improved; as resources can be distributed in greater proportion to the better performing groups (Antoniadis and Grand 2007). Otherwise, it will be difficult to shield the cooperators from the defectors who rarely or never share their resources.

For easy understanding, we differentiate our current system from our previous work (Purvis et al. 2006). First we explain the results from the earlier system (Purvis et al. 2006) for comparative purposes. In that work, all the 5 groups started with a similar number of cooperators in each group. Later the groups were separated into 2 groups having most of the cooperators, 2 groups having most of the non-cooperators and the middle group having a mixed population of both. But that earlier work employed localized group monitors and was therefore less scalable and semi-centralised.

The work presented in (Savarimuthu et al. 2009) is based on a closed society but cannot be applied to systems that are open and distributed. Even though the mechanism achieves self-organisation, it is suitable for systems in which the performances of the other groups are directly revealed to the agents in the society.

The work presented in (Savarimuthu et al. 2010) shows the self-organisation of groups using similar mechanisms and it has been improved upon in this current work. The differences between the work presented in (Savarimuthu et al. 2010) and current work are as follows. In earlier work (Savarimuthu et al. 2010) the game was played for certain iterations and the gossip information was stored. Later the agents use the stored gossip information when they play. In the current setup, the agents start using the gossip right from the start. If there is no information the agent is considered as a new player and allowed to play or enter into any group. As they play, the gossip is also stored and used. In the earlier work wealth has been taken into account. If the wealth of an agent has not increased in the last certain number of iterations then the agent decides to move. In the current setup, instead of wealth if the rejection limit is met then the agent decides to move. We found that using a rejection limit works better for group separation than basing the decision on wealth, since it is likely that the wealth will increase for a certain number of iterations (because the agents play with bad agents if the gossip information was not available, hence the wealth of the bad players might increase).

In the earlier work (Savarimuthu et al. 2010) new players are introduced into the lowest group in the society and they are expected to build their way up to the higher groups based on their behaviour (cooperativeness). For that it was necessary to keep track of the lowest group of the system all the time, which is not a recommended practice if we want to achieve a decentralised environment. In the current setup new agents go to random groups in the society. As they are new they have no past behaviour to track and they are allowed in any group as they come in. Eventually they will end up in a group based on their behaviour by the mechanism we have in place. In the earlier work the remaining agents in a dismantling group go to the lowest performing group. In the current setup, they can apply to other groups and go to the group that accepts them. If they are not allowed then they keep trying to get entry into one of the groups.

In summary, our current work focuses on addressing the free-riding problem in an open, dynamic and distributed society. The current work supports dynamical formation of groups of varying sizes and self-organisation based on cooperativeness.The work presented here provides an improved model when compared to the model presented in (Savarimuthu et al. 2010).

Future Work

The research area of cooperation in distributed societies of autonomous software agents is wide and there are many aspects that were outside the scope of the present work.

- Consideration of agents changing behaviour: In this work we have not considered agents changing their behaviour in their life time. In real life, bad agents may redeem themselves or may be forced to cooperate through institutional punishment mechanisms. In our future work, we intend to examine more advanced situations in which agents can dynamically alter their cooperation strategies. That would mean that an agent could start with a certain cooperative value, but later based on the circumstances (e.g.based on learning), decide to change its behaviour. For example, an agent could try to enhance its performance by becoming a "bad guy" temporarily and then returning to being a "good guy", since it may estimate that its potential rewards could be even higher because of occasional cheating in a good group of agents. Such behaviour changing mechanisms can be investigated in the future.
- Varying gossip type: In the future, we intend to examine the types of gossip in the system to determine conditions under which the gossip mechanism leads to separation of groups and conditions under which it does not lead to the separation (i.e. by experimenting with different types of gossip (e.g. about other groups, about other agents' trustworthiness in providing gossip information)).
- Handling the lying problem: The systems investigated in this work make use of recommendations from other agents to decide whether to interact with another agent or not and also to know the performances of other groups, relying on the fact that the other group agents are honest in revealing the information about their group (i.e. these agents do not lie). However, this may not be the case in general. Agents being autonomous (and intelligently self-interested) may not want to share their group information (e.g. cooperativeness of the group) with outsiders and, worse still, may lie when such information is requested. This behaviour may lead to an undesired state of affairs in a society which may also have deleterious effect in segregation of groups. Our intention is to examine these issues in our future investigations. Another interesting avenue for research is to investigate additional mechanisms for handling these types of misbehaviours in agent societies.

Summary

We have presented a gossip based decentralised mechanism to facilitate the self-organisation of agent groups in open agent societies. Our mechanism helps in the separation of better performing groups from not-so-good groups. This also reduces the likelihood of bad agents exploiting the good agents in the better groups.Through agent based simulation we have demonstrated that our mechanism helps the sharing agents (cooperators) to move to better groups while the non-sharing agents are restricted from getting into the better groups. Thus, the mechanism achieves the separation of groups based on their cooperativeness. The mechanism allows for dynamic group formation through the splitting and dismantling processes. We have also demonstrated that our system is scalable. We have experimented with different gossip levels and false gossips and their impact in the system behaviour. Finally, we have compared our results with previous works.

We have shown how self-organisation of groups can be achieved in an open agent society that is scalable (i.e. the number of agents can be large), dynamic (i.e.agents can come and go), and employs decentralised notions for monitoring and control (i.e. social mechanisms (gossip and ostracism) that are decentralised). These properties make the system suitable for deployment in contemporary electronic societies such as P2P societies. We have also discussed the future extensions to the work carried out.

Declaration

This work has appeared (with results in Experiment 1, 2 and 3) in the 13th International Conference on Principles and Practice of Multi-Agent Systems published by Springer (Savarimuthu et al. 2010).

References

Antoniadis, P. & Grand, B. L. (2007). Incentives for resource sharing in self-organized communities: From economics to social psychology. In *ICDIM* (pp. 756–761). IEEE.

Boyd, S., Ghosh, A., Prabhakar, B., & Shah, D. (2006) Randomized gossip algorithms. *IEEE Transactions on Information Theory, 52*(6), 2508–2530.

Conte, R., & Paolucci, M. (2002). *Reputation in artificial societies: social beliefs for social order*. Norwell, MA, USA: Kluwer Academic Publishers.

de Pinninck, A. P., Sierra, C., Schorlemmer, W. M. (2008). Distributed norm enforcement: Ostracism in open multi-agent systems. In: P. Casanovas, G. Sartor, N. Casellas, R. Rubino (Eds.) *Computable Models of the law, languages, dialogues, games, ontologies, lecture notes in computer science, vol.* 4884 (pp. 275–290). Springer.

Dasgupta, P. (2003). Incentive driven node discovery in a p2p network using mobile intelligent agents. In H. R. Arabnia, R. Joshua, Y. Mun (Eds.) *Proceedings of the international conference on artificial intelligence, IC-AI '03*, vol. 2 (pp. 750–756), June 23–26. Las Vegas, Nevada, USA: CSREA Press.

Dunbar, R. (1996). *Grooming, gossip and the evolution of language*. London: Faber and Faber.

Dunbar, R. (2004). Gossip in evolutionary perspective. *Review of General Psychology, 8*(2), 100–110.

Eugster, P., Felber. P., & Le Fessant. F. (2007). The "art" of programming gossip-based systems. *SIGOPS Operating Systems Review, 41*(5), 37–42.

Ganesh, A. J., Kermarrec, A. M., & Massoulie, L. (2003). Peer-to-peer membership managementfor gossip-based protocols. *IEEE Transactions on Computers, 52*, 139–149. doi:10.1109/TC.2003.1176982. URL http://portal.acm.org/citation.cfm?id=642778.642782.

Gorodetsky, V., Karsaev, O., Samoylov, V., & Serebryakov, S. (2007). Multi-agent peer-to-peer intrusion detection. In V. Gorodetsky, Kotenko, I., Skormin V. A. (Eds.) *Computer network security, communications in computer and information science*, vol. 1 (pp. 260–271). Berlin, Heidelberg: Springer.

Hales, D. (2004). Self-organising, open and cooperative P2P Societies—From tags to networks. In S. Brueckner, G. D. M. Serugendo, A. Karageorgos, R. Nagpal (Eds.) *Engineering self-organising systems, Vol. 3464 of Lecture Notes in Computer Science* (pp. 123–137). Springer (2004).

Hardin, G. (1968). The tragedy of the commons. *Science, 162*, 1243–1248.

Hirshleifer, D., & Rasmusen, E. (1989). Cooperation in a repeated prisoners' dilemma with ostracism. *Journal of Economic Behavior and Organization, 12*(1), 87106.

Jelasity, M., Montresor, A., & Babaoglu, O. (2004). Detection and removal of malicious peers in gossip-based protocols. In *FuDiCo II: S.O.S.* Bertinoro, Italy.

Jelasity, M., Montresor, A., & Babaoglu, O. (2005). Gossip-based aggregation in large dynamic networks. *ACM Transactions on Computer Systems, 23*, 219–252. doi:10.1145/1082469.1082470.

Kempe, D., Dobra, A., & Gehrke, J. (2003). Gossip-based computation of aggregate information. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, FOCS '03* (pp. 482–491). Washington, DC, USA: IEEE Computer Society. URL http://portal.acm.org/citation.cfm?id=946243.946317.

Khan, S. K. A., & Tokarchuk, L. N. (2003). Interest-based self organization in group-structured p2p networks. In: *Proceedings of the 6th IEEE conference on consumer communications and networking conference, CCNC'09* (pp. 1237–1241). Piscataway, NJ, USA: IEEE Press. URL http://portal.acm.org/citation.cfm?id=1700527.1700827.

Mordacchini, M., Baraglia, R., Dazzi, P., & Ricci, L. (2010). A p2p recommender system based on gossip overlays (prego). In *Proceedings of the 2010 10th IEEE international conference on computer and information technology, CIT '10* (pp. 83–90). Washington, DC, USA: IEEE Computer Society. doi:10.1109/CIT.2010.502.

Nakamaru, M., & Kawat, M. (2004). Evolution of rumors that discriminate lying defectors. *Evolutionary Ecology Research, 6*(2), 261–283.

Nowak, M., & Sigmund, K. (1998). Evolution of indirect reciprocity by image scoring. *Nature, 393*(6685), 573–577. doi:10.1038/31225

Nowak, M. (2006). Five rules for the evolution of cooperation. *Science, 314*(5805),1560–1563.

Nowak, M., & Highfield, R. (2012). *SuperCooperators*. Canongate Books. URL http://books.google.co.nz/books?id=E2KcygAACAAJ.

Oh, J. C. (1999). Ostracism for improving cooperation in the iterated prisoner's dilemma game. In *Proceedings of the 1999 congress on evolutionary computation, CEC 99*, vol. 2 (pp. 891–896). doi:10.1109/CEC.1999.782517.

Ohtsuki, H., & Iwasa, Y. (2004). How should we define goodness?-reputation dynamics in indirect reciprocity. *Journal of Theoretical Biology, 231*, 107–120

Pacheco, J. M., Santos, F. C., Chalub, F. A. C. C. (2006). Stern-judging: A simple, successful norm which promotes cooperation under indirect reciprocity. *PLoS Computational Biology, 2*(12).

Paine, R. (1967). What is gossip about? an alternative hypothesis. *Man, 2*(2), 278–285. URL http://www.jstor.org/stable/2799493.

Paolucci, M., Marsero, M., & Conte, R. (2000). What is the use of gossip? A sensitivity analysis of the spreading of respectful reputation. In *Tools and Techniques for Social Science Simulation* (pp. 302–314). Physica-Verlag.

Purvis, M. K., Savarimuthu, S. De Oliveira, M., Purvis, M. A. (2006). Mechanisms for Cooperative Behaviour in Agent Institution, In T. Nishida, M. Klusch, K. Sycara, M. Yokoo, J. Liu, B. Wah, W. Cheung, Y.-M. Cheung (Eds.) *Proceedings of IEEE/WIC/ACM international conference on intelligent agent technology (IAT 2006)* (pp. 121–124), ISBN 0-7695-2748-5. IEEE Press: Los Alamitos, CA.

Roberts, J. M. (1964). The self-management of cultures. In: W. H. Goodenough (Ed.) *Explorations in cultural anthropology essays in honor of George Peter Murdock* (pp. 433–454). NewYork: McGraw-Hill.

Saroiu, S., Gummadi, P., & Gribbe, S. (2002). *A measurement study of peer-to-peer file-sharing systems*. Technical report UW-CSE-01-06002, University of Washington.

Savarimuthu, S., Purvis, M. A., & Purvis, M. K. (2009). Self-organization of peers in agent societies. *In IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology* (Milan, Italy, September 15–18, 2009). ISBN 978-0-7695-3801-3, Vol. 2 (pp. 74–77). Los Alamitos, CA, USA.

Savarimuthu, S., Purvis, M. A., Purvis, M. K. & Savarimuthu, B. T. R. Mechanisms for the self-organization of peer groups in agent societies. In *Proceedings of the of the 11th International Workshop on Multi-Agent-Based Simulation (MABS2010)* (pp. 91–104). Toronto, Canada, May (2010).

Savarimuthu, S., Purvis, M. K.,Purvis, M. A., & Savarimuthu, B. T. R. (2010). Gossip-based self-organising open agent societies. In *The 13th international conference on principles and practice of multi-agent systems, vol. 7057 of Lecture Notes in Computer Science* (pp. 105–120). Springer, November (2010).

Savarimuthu, S. (2010d) *Self-organising groups (gui for open society)*. University of Otago. URL http://unitube.otago.ac.nz/view?m=HbOw1pni7qS February 2010d.

Skyrms, B., & Pemantle, R. (2000). A dynamic model of social network formation. *Proceedings of the National Academy of Sciences, 97*(16), 9340–9346.

Skyrms, B. (2009). Groups and networks Their role in the evolution of cooperation. In: S. A. Levin (Ed.) *Games Groups and the Global Good, Springer Series in Game Theory* (pp. 105–114). Berlin, Heidelberg: Springer Physica-Verlag.

Smead, R. (2010). Indirect reciprocity and the evolution of moral signals. *Biology and Philosophy, 25*(1), 33–51.

Sommerfeld, R. D. , Krambeck, H. J., Semmann, D., & Milinski, M. (2007). *Gossip as an alternative for direct observation in games of indirect reciprocity proceedings of the national academy of sciences of the United States of America*, Vol. 104, No. 44 (Oct. 30, 2007) (pp. 17435–17440), Published by: National Academy of Sciences, Stable URL: http://www.jstor.org/stable/25450253.

Rebecca, S. B. (1956). *Some psychological mechanisms operative in Gossip, social forces*, Vol. 34, No. 3 (Mar., 1956) (pp. 262–267), Published by: University of North Carolina Press, Stable URL: http://www.jstor.org/stable/2574050

Suls, J. M. (1977). Gossip as social comparison. *Journal of Communication, 27*(1), 164–168. doi:10.1111/j.1460-2466.1977.tb01812.x.

Thomsen, R. (1972). *The origins of ostracism, A synthesis*, (Gyldendal: Copenhagen, 1972).

Yu, B., & Singh, M. P. (2000). A social mechanism of reputation management in electronic communities. In: M. Klusch, L. Kerschberg (eds.) *Proceedings of the 4th International Workshop on Cooperative Information Agents IV, The Future of Information Agents in Cyberspace, Lecture Notes in Computer Science* (pp. 154–165). London, UK: Springer. URL http://portal.acm.org/citation.cfm?id=647785.735555.

Zaharia, M. A., & Keshav, S. (2008). Gossip-based search selection in hybrid peer-to-peer networks. *Concurrency and Computation: Practice and Experience, 20*(2), 139–153. URL http://dx.doi.org/10.1002/cpe.1188