Mechanisms for the Self-organization of Peer Groups in Agent Societies

Sharmila Savarimuthu, Maryam Purvis, Martin Purvis,
and Bastin Tony Roy Savarimuthu

Department of Information Science, University of Otago,
Dunedin, New Zealand
{sharmilas,tehrany,mpurvis,tonyr}infoscience.otago.ac.nz

Abstract. New mechanisms for group self-organization in agent societies are investigated and examined in the context of sharing digital goods. Specifically we illustrate how cooperative sharers and uncooperative free riders can be placed in different groups of an electronic society in a decentralized manner. We have simulated a decentralized, open P2P system which self-organizes itself to avoid cooperative sharers being exploited by uncooperative free riders. Inspired by human society, we use social mechanisms such as tags, gossip and ostracism. This approach encourages sharers to move to better groups and restricts free riders without necessitating centralized control, which makes the system appropriate for current open P2P systems.

Keywords: Multi-agent Based Simulation, Cooperation, Sharing behavior and Artificial Societies.

1 Introduction

One of the common problems in a P2P network is that of free riders. In our context, free riders are those agents or nodes that do not contribute to the collective goals of the network society, but make use of the resources of the network [9]. These free riders decrease the overall performance of the society by contributing to the degradation of the common good [4] without contributing to the community. In a way, they can be considered as parasites.

Many existing approaches to social regulations in ecommerce employ a centralized mechanism to control free riders, whereby the system eliminates antisocial behaviour by employing a monitoring or governor agent [6, 9]. But these centralized mechanisms are computationally expensive for a system and can represent a bottleneck. Centralized control systems need a central manager to monitor and carry out punishment or to provide an incentive mechanism, which is not suitable for decentralized systems, due to the explosion of state spaces. In an open system it is inappropriate to rely on a centralized monitoring authority that monitors all possible state spaces that an agent can be in. Scalable modern P2P systems are entirely decentralized and open; hence to deal with this dynamic nature of digital societies, there is a need for a decentralized solution for dealing with the free-riding problem.

T. Bosse, A. Geller, and C.M. Jonker (Eds.): MABS 2010, LNAI 6532, pp. 93–107, 2011.

In this work we propose a decentralized solution that makes use of social mechanisms such as tagging, gossip and ostracism. 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. In this work, inspired by human society, we propose a new mechanism for agents to self-organize themselves into different groups, based on their behavior and in a decentralized manner.

2 Background and Related Work

Previous research has shown that tags can improve cooperation among participants and can induce "altruistic" behavior [3, 7]. For example, some researchers have shown that tag-based mechanisms have been successful in the evolution of cooperation using the Iterated Prisoner's Dilemma/ Prisoner's Dilemma (IPD/PD) games [3, 8].

By playing the "donation game", agents employing tagging achieved altruism in the model described by Riolo, Cohen and Axelrod [7]. In their model, tag and tolerance values are used to form groups, and an agent donates to another when the difference between its tag values is within the agent's tolerance level. Also an agent in their system could be a member of more than one group. In that case, that agent may donate to or receive from the group members of all those groups. This mechanism has been shown to achieve altruism among peers by making use of tags.

The knowledge sharing game is about sharing knowledge (information/skill) within a society that is composed of sharers and non-sharers. In the context of the knowledge sharing game [10-12], it is shown that tagging can help to increase sharing behaviour. The work presented in [10] describes the effect of tag-based mechanisms for sustaining knowledge through sharing behavior, and it describes the conditions under which sharing behavior spreads through the society and how knowledge is shared and sustained in the society.

In the work of Purvis, Savarimuthu, Oliveira and Purvis [6], the cooperative self-organization of peers in different groups was achieved in playing the PD game, 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 or 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. But this approach was still semi-centralized, because it required a local monitoring agent for each group. In addition it was a closed system model, so it did not fully support open, distributed P2P systems.

In Hales's work [3], the PD game was played in a P2P network among nodes. This work extends his previous work on tags, to networks, where a 'neighbor list of nodes' is considered to be a tag and '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.

In our present work, instead of the PD game, we have adopted the more practical scenario of sharing digital goods in electronic societies. We investigate how we can achieve the separation or self-organization of groups based on their behavior in a decentralized manner and 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 and restrict their entree to better groups where the access to resources are better and hence the quality of service/performance is higher. By doing so, the performance of the whole system can be improved, because resources can be distributed in greater proportion to the better performing groups. 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 system from the system of Purvis, Savarimuthu, Oliveira and Purvis [6], see Table 1.

Table 1. Differences between the earlier system and our present system

Earlier system	**Present system**
Semi-centralised	Decentralised
Used monitoring agents	No monitoring, distributed
Used voting mechanism	Used gossip mechanism
Closed system	Open system

3 Experimental Model

Our experimental model presents a social situation in which the agents have the option to share or not share. Sharing costs the donor who shares. But the receiver receives the benefit (b) without incurring any cost (c). The sharer who shares the file loses -1 as a cost. The receiver who receives the file gets 2 as a benefit. Non-sharing is the selfish option which benefits the individual but is not good for the society. Sharing leads to the betterment of the society as opposite to the non-sharing which deteriorates the common good. Everyone will be better off if everyone shares. 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" [4]. Some of the model properties and mechanisms used in our experiments are described below:

3.1 Tag Groups

The tags we use are simply markings that are "visible" to other agents and are employed for grouping purposes. Some natural biological tagging examples are birds flocking together, animals forming herds, and ants forming a colony. They interact within their tagged group – they act together, and those small interactions among them can lead to the emergence of collective behavior. Thus the tagging mechanism that we use is inspired by nature, and it has been widely used to model the behaviour of artificial agent societies. A simple way to think of these tags is to assume that they represent group identifiers for sets of agents: agents having the same tags belong to the same group. Tagging is thus a straightforward and lightweight approach for facilitating cooperation [3].

3.2 Gossip Mechanism

Gossip has long been an effective mechanism for passing information in human society [2, 14]. Similarly in agent societies, agents use gossip to learn about other agents [5]. This mechanism maintains partial (i.e. not complete) information about agents in the society in a distributed and scalable fashion within the system. This gossip mechanism can be considered as 'distributed referral', and it is described more fully in the experimental setup.

3.3 Social Ostracism

Agents will refuse to interact (share resources) with another agent if that other agent is identified as the "worst", i.e. the least cooperative agent in the group. If other agents are not interacting with the worst agent, the worst agent will eventually choose to leave the group on its own, since it no longer has opportunities to increase its wealth. This is a kind of 'ostracism' [1, 15].

3.4 Agent Attributes

In experiments described in this paper, the agents have fixed, randomly assigned attribute values which represent how they behave. One agent attribute concerns cooperation: agents have a randomly assigned cooperation value between 0 and 10 that represents how much they cooperate (share), with 0 representing maximally uncooperative and 10 representing maximally cooperative. This value is known as the cooperativeness of the agent. Agents with low cooperativeness, hardly or never share. These agents are known as free riders. Agents also 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, and 10 identifies the most tolerant.

4 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. In the initial setup 100 agents are put into five random groups. The group is called a "tag group" which 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. Each agent has a gossip blackboard to store the gossip messages from other agents of its group. Each agent also has a memory of any previous groups to which it has belonged. Each agent is initialized with a random cooperative value and a random tolerance value. The experiment was executed for 5000 iterations. The procedure of the experiment is explained in detail below.

4.1 Gossiping

In every iteration, a certain random percentage of the 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 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, 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. The first 500 iterations (out of 5000) are played in this manner to build up a distributed gossip repository among the players. For further illustration, the operation of how peers publish gossip is outlined schematically in Algorithm 1. In the scheme illustrated in Algorithm 1, there are three players A, B and C, belonging to the same group. A is the taking-player, B is the giving-player, and C is the gossip holder.

Algorithm 1. To publish gossip

```
begin
    A requests for file to B ;
    if B shares then
        A gossips good about B to C;
    else
        A gossips bad about B to C ;
    end
end
```

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.

After 500 iterations, the agents begin using the received gossip information to decide whether or not to play with a taking-player. When a player requests a file, the giving player can check with five other random agents (asking them what they know from the gossip information they have received) whether this asking agent is the worst cooperator of their group. The worst player is the one who has been uncooperative most of the 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. Otherwise this giving player interacts (sharing a file or not based on its own cooperativeness). The operation of how peers use gossip is outlined in Algorithm 2, where B and D are the players in the group. Assume here that *B* is the taking-player, *D* is the giving-player, and *D* collects the gossip information from any 5 other agents and checks whether *B* is the worst player (from the available gossip information) .

Algorithm 2. To use gossip within group

```
begin
    D requests 5 other players for gossip about B;
    D receives gossip;
    if B is the worst player then
        D refuses to play with B;
    else
        D plays with B;
    end
end
```

When only a few agents (less than 5) have gossip about a taking-player, then only the available information is taken into consideration. Sometimes it can be the case that none of the players has gossip about the taking-player. In such a case the taking-player is considered not to be the worst player, a privilege similar to what happens when a new player joins a group.

4.2 Leaving a Group

A player can leave a group if its tolerance level is surpassed or when its wealth has not increased recently. We call this 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 (breaching the agent's tolerance limit) the agent will decide to leave that group and move to another group.

In addition, making use of gossip information, the agents will stop playing with the worst player in their group. Every time they play, the giving-players check with other agents about the taking-players and check whether that player is considered to be the most uncooperative. If an agent is regularly rejected from play, then, of course, that agent's score will not increase. If, over a given period of play opportunities (here, 15 iterations), an agent's wealth has not increased, then it will choose to leave that group and move to another group. Since the other players in its current group are not playing with it, it will be better off moving to another group, irrespective of that group's co-operativeness/performance. Thus the worst player leaves the group on its own accord, without any control applied on it.

4.3 Joining a Group

The hopping peer then collects information about other groups from their group members. Then it decides to which group to request admission. 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 the Table 2 explains 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 1200th iteration. Since the composition of groups invariable change over time, the cooperativeness of any group will change as time progresses. 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 get to see which agent has moved into this group recently from other groups. Taking into consideration the most recent information available, the agent decides where to move based on this information. For example assuming the current iteration is 1400, the latest information collected from the group members is given in Table 3.

Table 2. Previous group history

Group No	Iteration No	Cooperativeness
1	560	4.5
3	700	6.0
2	1200	6.4

Table 3. Latest available information

Group No	Iteration No	Cooperativeness
5	1330	8.1
3	1170	7.5
2	1200	6.4
1	1199	3.8

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 don't allow entry into their groups, 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 – it is being shunned by the other members for being the worst member of its group. Because of play rejections, its wealth will not advance, and it will want to leave and find some other group in which it can find players to play with. If the better groups do not allow entry, the agent will go to lower and lower groups, since it is better off moving to any new 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.

4.4 Calculating the Entry Value

The hopping peer asks any randomly chosen agent in the group to which it seeks entry for 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 agents whose cooperativeness values are greater than or equal to a value calculated by a formula (given below). 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:

$$\mathbf{EV = AC - (C1 / (SL - S)^{C2}) + C3^{(S-SU)}}$$

The group Entry Value (EV) is calculated considering the given group's Average Cooperativeness (AC) and its group Size (S). AC is the average cooperativeness of the group calculated through the gossip mechanism, and S is the size of the group. C1, C2, C3 are constants whose values in our experiments are 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). It is inappropriate or inefficient for groups of players or traders to become too big or too small. In our experiments, SU was set to be 25, and SL was set to be 10. That means if the size of the group is 10 or below the entry qualification value is set at 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 (AC) 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. This can be identified in Figure 1 by examining the line Avg6 for size 14 and for size 25.

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. 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. In that case the hopping peer will try to enter into other groups. This process is outlined in Algorithm 3. The hopping peer will ultimately get into a group where its cooperativeness is eligible to enter. If no such group is available, the hopping peer stays in its current group.

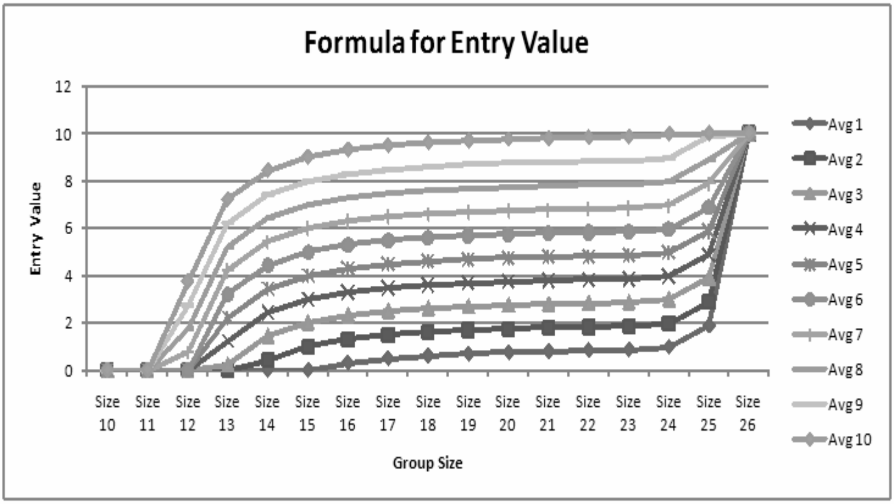

Fig. 1. Entry value calculated by the formula

Algorithm 3. To use gossip between groups

begin
E requests 5 other players for gossip about F;
E receives gossip about Fs estimated cooperativeness;
E calculates entry value;
if *Fs estimated cooperativeness* $\geq$ *entry value* **then**
F gets entry;
else
F does not get entry;
end
end

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. The overall process is outlined in Algorithm 4. A demo video can be seen in this link [13].

4.5 Results and Comparison

To consider the overall performance of our mechanisms, we first explain the results from the earlier system [6] for comparative purposes. In that earlier result, all the 5

Algorithm 4. Pseudocode for the overall process

```
begin
    initialization;
    bootstrap agents in groups;
    foreach iteration do
        select random number of agents;
        if iteration is less than 500 then
            foreach selected agent do
                play with another agent in the group;
                collect payoff;
                gossip;                                          // Algorithm 1
            end
        else
            foreach selected agent do
                play with another agent based on gossip;         // Algorithm 2
                collect payoff;
                gossip;                                          // Algorithm 1
                if selected agent's tolerance level is met then
                    selected agent leaves the group;
                    joins another group under certain condition; // Algorithm 3
                end
                if selected agent's wealth has not increased then
                    selected agent leaves the group;
                    joins another group under certain condition; // Algorithm 3
                end
            end
        end
    end
end
```

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 thatearlier work employed localized group monitors and was therefore less scalable and distributed.

We present our results here using the decentralized approach in Figure 2. Initially all the five groups were randomly seeded and started with roughly similar average cooperativeness values among their members. They ended up showing a separation among the groups with respect to their cooperativeness values. Group4, with mostly the best cooperators, Groups 2 and 3, with mostly non-cooperators, Groups 1 and 5, with moderate ones. This partitioning was achieved using social mechanisms without central control.

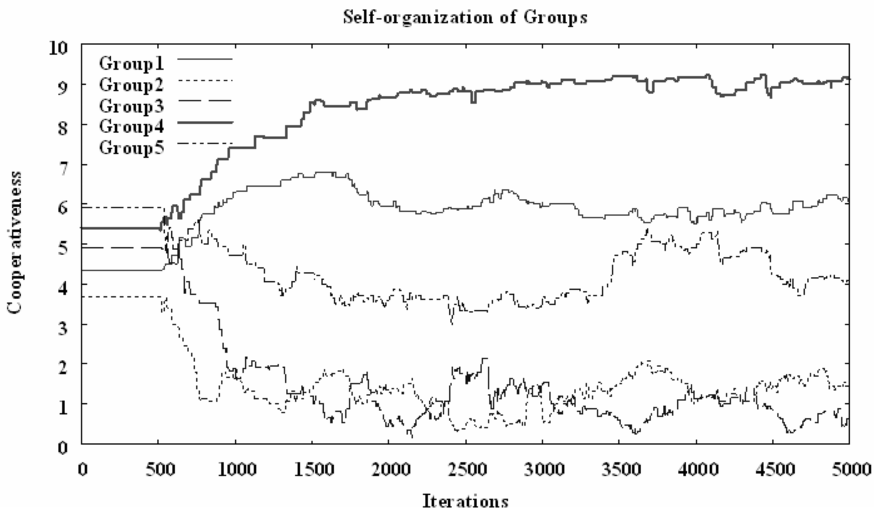

Fig. 2. Self-organization of groups based on cooperativeness in closed society

A paired-samples t-test was conducted to compare the separation of groups based on cooperativeness (standard deviation)) at the start and end of the runs. The paired t-test was performed with null hypothesis for 30 sample runs. The standard deviation of groups' cooperativeness at the start and end of the run were measured. There was a significant difference in the values at start (M=0.71, SD=0.25) and at end (M=3.27, SD=0.31) conditions. The average difference between the mean values (M=2.55, SD=0.07, N=30) was significantly greater than zero, t=33.77, two-tail p=($7.78*10^{-25}$), providing evidence that our mechanism is effective in producing the separation of groups based on cooperativeness with a 95% confidence interval.

We also experimented by varying the number of agents that are contacted to collect gossip. The number of agents from which gossip was provided varied between 2, 5 and 15. From the 30 sample runs collected, the mean of the standard deviations of groups, for gossip size 2, 5 and 15 are 2.80, 3.22 and 3.23 respectively. We noticed significant difference when we compared 2 with 5. Collecting gossip from 5 agents has resulted in better separation than from 2 agents. But when we compared 5 with 15 there is 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.

4.6 Adding Openness

Our aim has been to develop a self-organizing open and dynamic system, where new agents may come into the society and also agents may leave the society at any time. New unknown peers are allowed to join the society by gaining entry into the lowest ranked group. They can build their way up to higher groups, based on how well they perform in the eyes of their peers. 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 decentralized manner without explicit control at the top level. Forming groups using tags is helpful, since it is scalable and robust. For higher numbers of peers, more tag groups can be formed, and that process will scale well for any number of peers. Now, in the new arrangement, agents are set to have lifespans, which 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. Thus we added openness to the society (agents can arrive and leave).

In our approach, a group splits into two if the size of group reaches 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 certain limit (5) then the group dismantles. The remaining agents in the group go to the lowest group.

4.7 Results and Discussion

The self-organization of groups in the open society is shown in figure 3. The results from a sample run for 5000 iterations are presented in figure 3. Out of 5 initial groups, Group 4 dismantled. Group 3, which is the most cooperative group, split into two most cooperative groups, group 6 and 7. Group 5 also split into two groups 8 and 9, of which Group 9 is the lowest group. Note that the new groups formed by splitting have some difference in their cooperativeness since the most cooperative ones from the splitting group team up to form one group, and the rest form the other group. This is an ongoing process, because further groups will be formed and dismantled based on the arrival and leaving rates of the agents.

To test the scalability, we ran the experiment for 10000 iterations, by having the initial population set to be 100 in 5 initial groups and also having hundreds of agents enter the scene randomly over subsequent periods. Agents also leave the society when their life span is over. This experiment has scaled well by forming 23 groups over 10000 iterations, in which 11 of them have dismantled or split and the others are operational groups by the end of the iterations. It shows the system can scale well for any number of agents just by forming or dismantling groups dynamically. In open societies like these, agents cannot have a global view of all the groups. They have a limited view which means they know about the groups where the agents and its group members have been before.

It can also be observed from the results that the system mechanisms lead to the filtering out of the worst peers, and these mechanisms restrict those uncooperative peers from gaining access to the good, more cooperative, groups. This also helps to

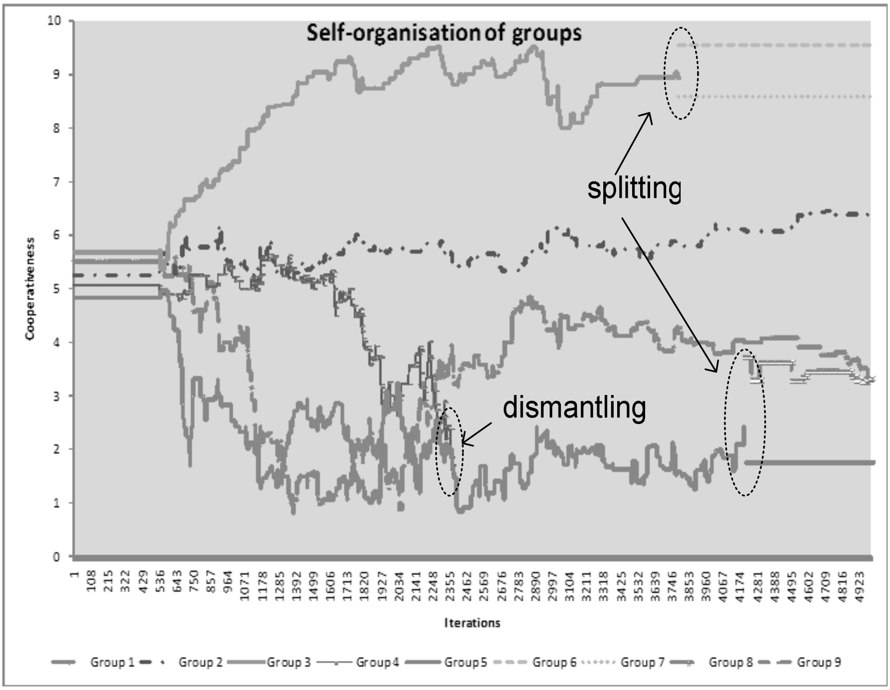

Fig. 3. Self-organization of groups based on cooperativeness in open society

improve the betterment of the society by protecting cooperators from not being exploited by free riders. The best players (most cooperative) get to access or/enter into any group. Thus the system has distributed operative mechanisms that lead to the following social principle for the individual agents: "the better you are the better your chances are".

Using the gossip mechanism, agents share their interaction experience with some other agents. Unlike most reputation mechanisms, where agents keep track of all the reputation information of other agents, our system distributes a subset of this information among the individuals in the group. This mechanism operates and converges to satisfactory results in the context of a partial-view of truth about the world states, which is a realistic and scalable feature of open agent societies. If any peer leaves a group, only a limited amount of information is lost, which ensures the robustness of the system. Peers might leave the society (because of bad behavior) and try to re-enter again. But our system filters out peers in separate groups, based on their behavior. It does not matter how many times an uncooperative agent would enter, it will inevitably end up in the worst group.

5 Conclusion and Future Work

The system presented in this paper is suitable for sharing digital goods in open P2P systems where the aim is to improve societal performance by avoiding free riding. It has produced self-organization, or the so called self-balancing of P2P systems, in a distributed and dynamic manner.

Our system takes advantage of social mechanisms, such as tagging to form groups, gossip to pass information, and ostracism to shun bad behavior. As a result, it shows the self-organization of groups based on behavior (cooperativeness). In our future work, we intend to examine more advanced situations in which peers dynamically alter their cooperation strategies. That would mean that a peer could start with a certain cooperative value, but later based on the circumstances, it could decide to change it accordingly. (It could try to enhance its performance by becoming a "bad guy" temporarily and then returning to being a "good guy").

In the future work we will also consider misbehavior of agents, by adding lying in the gossip mechanism.

References

1. de Pinninck, A.P., Bas, S.C., Schorlemmer, M.: Distributed Norm Enforcement: Ostracism in Open Multi-Agent Systems. In: Casanovas, P., Sartor, G., Casellas, N., Rubino, R. (eds.) Computable Models of the Law. LNCS (LNAI), vol. 4884, pp. 275–290. Springer, Heidelberg (2008)
2. Eugster, P., Felber, P., Le Fessant, F.: The "art" of programming gossip-based systems. ACM SIGOPS Operating Systems Review-Special topic: Gossip-Based Networking 41(5), 37–42 (2007)
3. Hales, D.: Self-Organising, Open and Cooperative P2P Societies - From Tags to Networks. In: Brueckner, S.A., Di Marzo Serugendo, G., Karageorgos, A., Nagpal, R. (eds.) ESOA 2005. LNCS (LNAI), vol. 3464, pp. 123–137. Springer, Heidelberg (2005)
4. Hardin, G.: The Tragedy of the Commons. Science 162, 1243–1248 (1968)
5. Jelasity, M., Montresor, A., Babaoglu, O.: Detection and removal of malicious peers in gossip-based protocols. In: FuDiCo II: S.O.S., Bertinoro, Italy (2004)
6. Purvis, M.K., Savarimuthu, S., De Oliveira, M., Purvis, M.A.: Mechanisms for Cooperative Behaviour in Agent Institution. In: Nishida, T., Klusch, M., Sycara, K., Yokoo, M., Liu, J., Wah, B., Cheung, W. (eds.) Proceedings of IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2006), pp. 121–124. IEEE Press, Los Alamitos (2006) ISBN 0-7695-2748-5
7. Riolo, R.L., Cohen, M.D., Axelrod, R.: Cooperation without Reciprocity. Nature 414, 441–443 (2001)
8. Riolo, R.L.: The Effects of Tag-Mediated Selection of Partners in Evolving Populations Playing the Iterated Prisoner's Dilemma. Santa Fe Institute (1997)
9. Saroiu, S., Gummadi, P., Gribbe, S.: A measurement study of peer-to-peer file-sharing systems, Technical report UW-CSE-01-06002, University of Washington (2002)
10. Savarimuthu, S., Purvis, M.A., Purvis, M.K.: Altruistic Sharing using Tags. In: Proceedings of the 6th International Workshop on Agents and Peer-to-Peer Computing, Estoril, Portugal (2008)
11. Savarimuthu, S., Purvis, M.A., Purvis, M.K.: Emergence of Sharing Behavior in a Multi-agent Society using Tags. In: Proceedings of IEEE/WIC/ACM International Conference on Intelligent Agent Technology, Sydney, Australia, pp. 527–530 (2008)

12. Savarimuthu, S., Purvis, M., Purvis, M.: Tag-based model for knowledge sharing in agent society. In: Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems, Budapest, Hungary, May 10-15, vol. 2 (2009); International Conference on Autonomous Agents. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, pp. 1299–1300 (2009)
13. Self-organising groups (closed society), Video link on youtube, http://www.youtube.com/watch?v=a_3MOfeUy2Y
14. Rebecca, S.B.: Some Psychological Mechanisms Operative in Gossip. Social Forces 34(3), 262–267 (1956), http://www.jstor.org/stable/2574050
15. Thomsen, R.: The Origins of Ostracism, A Synthesis (Gyldendal: Copenhagen, 1972) (1972)