Comput Math Organ Theory (2012) 18:419–444
DOI 10.1007/s10588-011-9105-7

SI: AGENT-DIRECTED SIMULATION

How behaviors spread in dynamic social networks

Yu Zhang · Yu Wu

Published online: 10 December 2011

Abstract In this paper, we explore how decentralized local interactions of autonomous agents in a network relate to collective behaviors. Earlier work in this area has modeled social networks with fixed agent relations. We instead focus on dynamic social networks in which agents can rationally adjust their neighborhoods based on their individual interests. We propose a new connection evaluation theory, the Highest Weighted Reward (HWR) rule: agents dynamically choose their neighbors in order to maximize their own utilities based on rewards from previous interactions. We prove that, in the two-action pure coordination game, our system would stabilize to a clustering state in which all relationships in the network are rewarded with an optimal payoff. Our experiments verify this theory and also reveal additional interesting patterns in the network.

Keywords Social networks · Dynamic networks · Emergence of social norms · Local social interactions · Pure coordination game

1 Introduction

Social simulation is a class of simulations used to study social behavior and phenomena (Axelrod 1997). With advances in modern technology, computer social simulation is becoming a more and more important tool in today's research. Large-scale simulation experiments, which might in the past have taken researchers months or

Y. Zhang (✉)
Department of Computer Science, Trinity University, San Antonio, TX 78216, USA
e-mail: yzhang@trinity.edu

Y. Wu
Department of Computer Science, Stanford University, Stanford, CA 94305, USA
e-mail: ywu2@stanford.edu

even years, can now be run within minutes. Computer simulation has not only saved researchers a great amount of time but also enabled them to study numerous macro topics that were impossible to study experimentally in the past. In particular, the multi-agent system (MAS) has emerged, facilitating the study of large-scale social networks.

Agent-based social simulation (Davidsson 2002) uses agents to model such social entities as people, groups and towns. The agents usually have limited intelligence and memory, so that they adjust their future behaviors based on their environment. The simulation has been run on various kinds of social networks (Dorogovtsev et al. 2008), which simulate various social structures (Newmann 2003). Much work has been done in this area and many results have gained attention, such as the six-degree separation theory derived from the small-world network (Watts 1999), the networks of movie actors, in which edges demarcate that two actors performed in the same movie (Watts 1999), the friendship networks of high school students (Fararo and Sunshine 1964), the web of human sexual contacts (Liljeros et al. 2001), and the scientific citation network (Redner 1998).

In reality, simulations using the MAS can be conducted in two ways: One is to build a model as realistic as possible for a specific scenario, such as the moving crowds in an airport (Turner and Penn 2002). The other is to build an abstract model that captures the essence of human activities and can be used to model many different social networks of the same kind. For instance, Kempe et al. (2003) provide a constant-factor approximation algorithm to test the manner in which an innovation spreads in a social network. Our research belongs to this latter category.

Researchers working on an abstract social model usually focus on the study of "generative social science" (Epstein 1999)—that is, how the decentralized local interactions of autonomous agents generate social norms. A *social norm* is defined as a regular behavior that is a solution to a recurrent or continuous social cooperation problem (Younger 2004; Dignum 1999; Kittok 1995; Lewis 1969). One basic question we might also ask about social norms is whether the agents in the social network will eventually agree on one action. Will the entire network come to one solution? In different networks and under different algorithms, results can vary drastically. The agents might converge into various patterns and at different times.

Generally speaking, two categories in this area study the emergence of social norms. The first category studies social norms in *static* networks (Jiang and Toru 2007). A network is called *static* when the edges are never created or removed after the generation of the graph. In this category, one of the most significant findings is by Shoham and Tennenholtz (1997), who proposed the Highest Cumulative Reward (HCR) rule. In each time step, an agent adopting the HCR rule switches to a new action if and only if the total rewards obtained from that action in the last time step are greater than the rewards obtained from the currently chosen action in the same time period. For instance, a customer would only switch to another frequent-flyer program when the rewards from an alternate program surpassed those offered by both her current provided and its other competitors. The authors simulated various networks under different parameter configurations (e.g., memory update frequency, memory restart frequency, and both) to experiment on an agent's effectiveness in learning about its environment. They also theoretically proved that under the HCR rule the network will always converge on one social norm.

Jordi Delgado's (2002) Generalized Simple Majority (GSM) rule is another important finding in this "static" category of social norms. Agents obeying the GSM rule will change to an alternative strategy if they have observed more instances of it in other agents than their present action. This rule generalizes the simple majority rule (Walker and Wooldridge 1995). As the randomness $\beta \to \infty$, the change of state will occur as soon as more than half of the agents are performing a different action. In this case, the GSM rule will behave exactly as the simple majority rule does.

The second category of research on agents' social behavior is the study of evolutionary or *dynamic* networks (e.g., Borenstein and Ruppin 2003; Zimmermann and Eguiluz 2005; Jin et al. 2001; Skyrms 2004). In a *dynamic network*, edges are created and removed as the network evolves. In today's world, many important virtual networks, such as e-commerce networks, SNS (Social Networking Service), etc., have fewer stable connections among agents and thus their network structures constantly change. Here researchers investigate the possible ways in which an agent population might converge on a particular strategy. Unfortunately, strong assumptions about the agents and their connections to nearby agents (a.k.a. their neighbors) usually make such experiments unrealistic. For example, agents often have no control over their neighborhood. Also, agents do not employ rational selfish reward-maximizing strategies but instead often imitate their neighbors. Finally, agents are often able to see the actions and rewards of their neighbors, which is unrealistic in many social settings.

To address the above problems, Zhang and Leezer (2009) developed the Highest Rewarding Neighborhood (HRN) network model, which enables agents to update their connections through a rational strategy. In this way, any static network can be easily transformed into a dynamic network through the HRN rule. The HRN rule allows agents to learn from their environment and compete in networks. Unlike the agents who can observe their neighbors' actions and imitate them (e.g., Borenstein and Ruppin 2003), HRN agents employ selfish reward-maximizing decision-making strategy and are able learn from their environment. Under the HRN rule, cooperative behavior emerges even though agents are selfish and attempt only to maximize their own utility, because agents are able to break unrewarding relationships and are thus able to maintain mutually beneficial relationships. This leads to a Pareto-optimum social convention, in which all connections in the network are rewarding.

This paper proposes a Highest Weighted Reward (HWR) rule, an extension of the HRN rule (Zhang and Leezer 2009). In the HRN rule, the agent values its neighbors based on the total rewards collected from them in the past, with all interactions being weighted equally throughout the agent's history. However, such an emphasis will cause the agent to focus too much on the history of its neighbor and fail to respond promptly to the neighbor's latest action. Motivated by this fact, we have developed the HWR rule. HWR allows the agents to use a discount factor to devalue past interactions with their neighbors. The more recent the interaction is, the more heavily the corresponding reward is weighted in the evaluation function. In our research, we identified certain patterns from the simulation results and then demonstrated the existence of the pattern both theoretically and empirically. Later, we compared the differences between our model and classic static network.

To our knowledge, we are one of the first groups to study the emergence of social norms under the self-interest setting in dynamic networks. The proposed model incorporates the individual agent decision model into networks with changed relations. We prove that in the two-action pure coordination game, in which players receive higher rewards when they cooperate and choose the same action, our system will stabilize to a clustering state in which all relationships in the network are rewarded with the optimal payoff. This research contributes in numerous ways to social-norms research. By employing agents who learn from their environment we provide a more complex setting in which to analyze the spread of behavior in social networks and the evolution of a social system. This work also contributes to the fields of Social Simulation by researching the emergence of social norms in networks of selfish agents.

The rest of the paper is organized as follows. Section 2 reviews the literature on existing models that lead to social norms emerging from a social network. Section 3 defines the HWR rule. Section 4 defines our argument regarding stable pattern of the proposed HWR rule. Section 5 is the proof that, in the two-action pure coordination games, our system will stabilize in a clustering state and at that time all relationships in the network are rewarded with the optimal payoff. Section 6 presents a series of experiments to evaluate the concept, the model and the manner in which our system stabilizes in a stable state. Section 7 offers a discussion of our findings, bringing together the proof presented in Sect. 5 with our experiments presented in Sect. 6. In the final section, we summarize our findings and offer a few possible avenues for future research.

2 Related work

Social networks have long been an important research topic in many areas. A deep understanding of the network can shed light upon many key issues in sociology and ecology studies. Especially in recent years, modern technology has enabled more and more virtual networks to be layered over existing physical networks. This change has made the study of networks even more important. However, in many cases, the scale of a given network is so large that no researcher can afford to manually collect data and carry out related studies. In other cases, it is often too difficult to track the development of a network, such as the evolution of an animal population over many generations. As a result, a science of social network simulation has emerged and is becoming a more and more convenient tool for network researchers. With the help of network simulation, researchers are now able to study networks of enormous scale through simulations that can be finished in days or even hours. Network simulation also enables researchers to test and verify earlier network hypotheses.

In recent years, researchers from different areas have begun to show more and more interest in network models. Coming from various backgrounds, they have framed network problems in very different ways and have attempted to solve them with very different approaches (Banks 2009). Though this diversity of methods has caused confusion and inefficiency, each approach has also had its own pros and cons. One might argue that the social network is simply too complex a problem, that it is nearly impossible to unify different approaches into one single area. As a result, interdisciplinary approaches are often a must for social network study. Such interdisciplinary approaches have led to significant achievements. For example, people have used statistical models to capture the group patterns of African elephants, successfully supporting an existing ecological hypothesis (Vance et al. 2009). While the "fission-fusion" societies can be viewed as a classic network-clustering problem, the combined efforts of both statisticians and biologists have offered a deeper and more significant understanding of elephants. Similar research has also been carried out in other areas, such as brain networks in medical science (Chen et al. 2009).

Because of the importance of network studies, researchers have developed various kinds of models. In most networks, nodes usually share simple common properties. Consequently, the development of network models usually focuses on the relations among nodes. Most network models can be represented by a certain distribution of edges among groups of nodes. Therefore, many mathematical models have been developed to capture the essence of a network through representation of the edges among nodes, often with significant findings (Hoff 2009; Handcock and Morris 2009). With the help of mathematical tools, researchers have managed to develop random samples of certain network types or to predict hidden information and complete real-world network models.

Some researchers have taken such work a step further and studied the network models with dynamic connections. Though much research has focused on classic networks, such as random networks, small-world networks and scale-free networks, connections in most of these models tend to be static. To be sure, the classic models have been proven very useful in many real-world models, though it is doubtful whether they are a good fit for many modern dynamic networks. One common feature many modern virtual networks share is the unstable connections among network nodes. People constantly add and drop friends and change their neighbors in virtual networks, such as SNS. Zhang and Leezer (2009) have suggested the use of dynamic connections to capture the evolutionary essence of modern networks.

When developing new network models, researchers are forced to make assumptions and use them as the cornerstone of their model. For example, Zhang and Leezer made the simple assumption that all agents in a network are selfish and that they try to maximize their interests by changing their neighbors. The influence of prior knowledge has also been carefully studied by researchers. Frantz et al. (2009) have argued that the accuracy and variance of collected data can diverge greatly in different network models. Prior knowledge about a network model, such as information about its structure or scale, can help us to better evaluate the quality of collected data.

Once a certain network model is built, its corresponding special properties determine how interesting and meaningful the model is. Researchers have studied a wide range of properties for network models and have set up corresponding theoretical frames (Ehrhardt et al. 2008). For example, researchers have taken a keen interest in the robustness of networks. After a certain network topology is found, researchers are often interested to know how long the structure can be sustained under various violations and attacks. The robustness of many classic network models, such as the small-world network and the power-law network, have been thoroughly tested by researchers (Centola 2009). Though the research on macro-level properties is meaningful, studies about micro-level agent interaction can also offer a deeper understanding of network models. Because a network is no more than a set of agents interacting with one another, agents' local decisions undoubtedly have a huge influence over the entire network. For example, researchers have shown that the agents' local decision heuristics have a strong connection with the topology of the network (Pujol et al. 2005a). As a result, similar research has been carried out around local agent interactions, providing a deeper understanding of a network's structure.

One of the most frequently discussed properties is the emergence of social conventions. Social conventions can be observed in many real network cases. The study of social conventions, for instance, can help to shed light on many real-world problems. For example, when viewing each node as a human being, and the states as their opinions, social conventions can be used to study the spread of rumors (Acemoglu et al. 2010). With different models or different parameters set up, people can further various cases of rumor spreading and come up an effective solution to stop it in real-world cases (Molchanov and Whitmeyer 2011). We can likewise study the spread of diseases and other similar topics using social conventions.

There are many key factors that influence the form of social conventions. One of the most significant ones is local decision heuristics. One common assumption is that agents only have limited knowledge about their environment. They only know, for instance, who their neighbors are and what actions they are currently adopting. This limitation has left researchers with an interesting question: whether a network can reach a social convention without the administration of a central authority. Researchers have tried to guarantee the emergence of social convention through different approaches. A common and effective approach is to make a simple and reasonable model for an agent's local decision heuristics and begin to infer the emergence of social conventions from that point (Shoham and Tennenholtz 1997). During this kind of inference, researchers have to choose their assumptions carefully. If the assumption is too strong, then the decision model may appear less realistic. On the other hand, if the assumption is too vague, it would be much more difficult to finish the inference.

Another approach to the study of social conventions is the investigation of network structures. One of the common properties researchers pay special attention to is the edge degree distribution. Given a certain simple game, researchers are able to evaluate the probability of the emergence of social conventions (Lopez-Pintado 2008; Jones 2008). Though in this approach the emergence of social conventions is not guaranteed, such flexibility allows us a deeper and closer analysis of many real-world cases. Without worrying about whether a social convention is reachable, people can focus on building a reasonable local decision model for agents before investigating the possibility of a social convention.

Even when a social convention has been reached, there are still many interesting properties of the convention that need to be further studied. For example, one interesting question may be the speed at which a social convention is reached (Delgado 2002). In the real world, a social convention that takes a million years to reach may have little practical meaning. Similarly, researchers could investigate how robust the social convention is; whether it exists in a final static state. or whether the agent ends up with an efficient action decision with optimal rewards. It appears in some cases that agents choose attractive actions rather than efficient actions in order to reach the social convention (Pujol et al. 2005b). Whether this state is desirable and how to reach/avoid such cases are both interesting related questions.

Researchers have also noticed the emergence of not one but multiple social conventions. Networks can reach Pareto-optimum in many ways, and multiple social conventions can usually guarantee maximum rewards too. When the agents in a network are polarized and form their own closed groups, local conventions might also lead to the global Pareto-optimum layout. It is also easy to discover many real-world examples of such network patterns. Human beings are likely to form opposing groups in many social areas, such as political parties, rival organizations, etc., and have been studied by researchers as the polarization of a network (Flache and Macy 2011).

3 Highest Weighted Reward (HWR) neighbor evaluation rule

3.1 The HWR rule

The HWR rule is based on a very simple phenomenon we experience in everyday life: human beings tend to value recent events more than those events that had happened further in the past. To capture this phenomenon, we introduce a time discount factor, which is usually smaller than 1, in order to linearly devalue the rewards collected from past interactions. When the time discount factor equals 1, the HWR rule will behave in the same way as the HRN rule. Notice that both the HWR and the HRN rules are neighbor evaluation rules—that is, the purpose of these rules is to help the agents value their connections with neighbors more wisely. In other words, these rules can only make decisions regarding connection choosing, but will not influence the agent's action when choosing decisions.

According to the HWR rule, an agent will maintain a relationship if and only if the weighted average reward earned from that relationship is no less than a specified percentage of the weighted average reward earned from every relationship. Let's carefully go through one cycle of the HWR evaluation step by step.

1. For each neighbor, we have a variable named *TotalReward* to store the weighted total reward from that neighbor. In each turn, we update the *TotalReward* through the following equation in which "c" represents the discount factor:

$$TotalReward = TotalReward \times c + RewardInThisTurn \tag{1}$$

2. In a similar way, we keep a variable named *GeneralTotalReward* to store the weighted total reward the agent receives from all neighbors in the history, and *TotalRewardInThisTurn* to store the total rewards collected from all neighbors. In each turn, we update the *GeneralTotalReward* through the following equation:

$$\begin{aligned} GeneralTotalReward = {} & GeneralTotalReward \times c \\ & + TotalRewardInThisTurn/NumOfNeighbors \end{aligned} \tag{2}$$

Here *TotalRewardInThisTurn* needs to be divided by the *NumOfNeighbors* in each turn in order to find the average reward per connection. Since we want to calculate the average reward of all interactions, the total reward in each round needs to be divided by the number of interactions in that turn.

3. When we start to choose non-rewarding neighbors, we calculate the average reward for every agent. The *AvgReward* is calculated in the following way: suppose the agent has been playing with the neighbor for n turns, then:

$$AvgReward = \frac{TotalReward}{1 + c + c^2 + \cdots + c^{n-1}} = \frac{TotalReward}{\frac{1-c^n}{1-c}} \tag{3}$$

4. Here we calculate the average total reward in a similar manner to step 3:

$$AvgTotalReward = \frac{GeneralTotalReward}{1 + c + c^2 + \cdots + c^{n-1}} = \frac{GeneralTotalReward}{\frac{1-c^n}{1-c}} \tag{4}$$

5. Finally, we compare the ratio of *AvgReward/AvgTotalReward* with threshold θ. If the former is greater than the latter, then we keep the neighbor; if not, we regard that agent as a bad neighbor. In this way, the agent can evaluate its neighbor with an emphasis on recent history. The rule is as follows:

$$\begin{cases} \text{if } \frac{AvgReward}{AvgTotalReward} > \theta, & \text{keep the neighbor} \\ \text{Otherwise,} & \text{disconnect bad neighbor} \end{cases} \tag{5}$$

The above explanation assumes an ideal environment in which agents have an infinite amount of memory for ease of exposition. In reality and in our experiments, agents have a limited amount of memory and can only remember their interactions for a certain number of turns.

3.2 Discount factor range

Generally speaking, the HWR rule could work with different games and in various networks. Because this rule only adds a dynamic factor, it does not contradict other aspects of the network. However, in some special cases, a certain experiment setup might make a discount factor value meaningless. Assume we are playing a pure coordination game (Erve and Roth 1998) with a payoff table (C: Cooperate; D: Defect) (see Table 1 in Sect. 6.1 for details about the game):

	C	D
C	1	0
D	0	1

Also assume the agents only want to keep the neighbors whose average reward is positive for them. In such a case, no matter how many rounds an agent has been playing with a neighbor, the agent's most recent reward will continue to dominate the whole reward sequence. To be more specific, if an agent receives a reward of 1 from a neighbor in their last interaction, but has constantly received -1 in the previous n trials, then the cumulative reward will be $1 - c - c^2 - \cdots - c^n$. If c is too low, the cumulative reward will still be larger than 0. We can find the range of c through the following derivation:

$$\begin{aligned} &1 - c - c^2 - \cdots - c^n > 0 \\ &\quad\Rightarrow \quad 1 > c + c^2 + \cdots + c^n \\ &\quad\Rightarrow \quad 1 > c\frac{1 + c^n}{1 - c} \end{aligned} \tag{6}$$

Since when $n \to \infty$, $c^n \to 0$, then

$$\Rightarrow \quad 1 > \frac{c}{1-c}$$
$$\Rightarrow \quad c < 0.5 \tag{7}$$

Based on this, when $c < 0.5$, the latest reward has a huge influence over the whole history. Therefore, we usually limit c in range of 0.5 to 1. In fact, for most cases c will take a value much higher than 0.5. Because the past rewards have been devalued exponentially, a small c value will cause the past rewards to be practically ignored very soon. In our common experiment setup, we let the agent have a memory size of 30. In this case, if $c = 0.5$, we see the oldest action be devalued by a factor of $0.5^{29} \approx 1.86\text{E–}9$, which makes the reward totally insignificant. In fact, even for a relatively high c value, such as 0.9, the last factor will still be as small as $0.9^{29} \approx 0.047$. Therefore, just as past rewards are devalued too heavily, we usually keep the c in the range of 0.8 to 1.

4 Pattern predictions

4.1 Argument

Based on the HWR rule, we make a few predictions about the outcome of the network pattern.

Argument Given a pure coordination game, placing no constraints on the initial choices of action by all agents, and assuming that all agents employ the HWR rule, then the following holds:

For every $\varepsilon > 0$ there exists a bounded number M such that if the system runs for M iterations then the probability that a stable clustering state will be reached is greater than $1 - \varepsilon$.

If a stable clustering state is reached, then all agents are guaranteed to receive optimal payoff from all connections.

By convention, a cluster refers to a set of users who are densely connected to each other. In order to further clarify the argument, the following definitions are given:

Stable Clustering State: A network reaches a stable clustering state when all the agents in the network belong to one and only one closed cluster.

Closed Cluster: A set of agents forms a closed cluster when no connection exists between any agent in the set and another agent outside of the set, when all the agents in the set are performing the same action.

Basically, what the argument claims is that the whole network will eventually converge into one single cluster of the same action or two clusters with different actions and that, once in such a state, the rewards collected from the network will be maximized.

5 Proof

There are some special networks that will never reach a stable end state. Researchers have proved under most circumstances that a static network will reach convergence when agents adopt the simple majority rule. However, during our research we discovered certain networks that would never converge. The most simple one can be referred as the "traffic light," which occurs in a network containing only two agents. At the beginning, the two agents are connected to each other and perform different actions. Because each agent has only the other agent as a neighbor, both will change their action in the next turn based on the simple majority rule. Apparently, these two agents will keep repeating this process and flipping between two actions. As a result, the network can never reach convergence. Even though this is a special case in a static network, dynamic networks feature the same problem. If our network is initialized as a complete network (though, theoretically speaking, it is unlikely that this situation would in fact happen), then the network will remain static, because it is impossible to add more connections between any pair of agents (because all agents are connected to each other). Accordingly, the special case we just pointed out is an extremely rare but valid case for our model.

Due to the existence of these special cases, not all networks will reach a stable clustering state. Inspired by Shoham and Tennenholtz's (1997) work, we therefore argue that as the time step approaches infinity, the probability of a network reaching the stable clustering state will also approach 1. In the next section, we show our argument in detail. In order to make our argument easier to understand, here we first explain its basic idea. The idea is very similar to the logic of the classic monkey and Shakespeare problem. Let a monkey randomly type on a keyboard. Given infinite time, the monkey will eventually finish a work of Shakespeare. In fact, this statement has been proved through computer simulation recently.[1] Our idea is similar: given a specific process that can happen with a very low probability and infinite time, the probability that this process can happen will approach 1. Our argument contains three major parts. First, a starting state that should be valid at any time step throughout the trial; second, an end state, which is also the goal state of the argument; third, a specific process that will lead the system from the starting state to the end state. In our argument, the starting state is a normal random state in the system. The goal state is the stable clustering state. We also describe the specific process in greater detail in our argument.

5.1 Definition

We have the following definitions.

- A_i denotes the ith agent in the network.
- a_i denotes the action choice for agent A_i. In coordination games, a_i has binary values representing different action choices.

[1] For details, please check the original article at http://www.jesse-anderson.com/2011/08/a-few-more-million-amazonian-monkeys/.

- R_{ij} denotes the link between agent A_i and agent A_j.
- N_i denotes agent A_i's neighborhood. If $A_j \in N_i$, then there exists a link R_{ij} between the two agents.
- C_i denotes agent A_i's coordinating neighbors. $C_i \subseteq N_i$. If $A_j \in C_i$, then $a_j = a_i$.
- $\overline{C_i}$ denotes agent A_i's non-coordinating neighbors. $\overline{C_i} \subseteq N_i$ and $\overline{C_i} \cap C_i = \phi$. If $A_j \in \overline{C_i}$, then $a_j \neq a_i$.
- We call agent A_i a *majority coordinating neighbor* if $|C_i| \geq \overline{C_i}|$; we call agent A_i a *majority non-coordinating neighbor* if $|C_i| < |\overline{C_i}|$.

Notice that any agent in the network must fall into one of the above two categories.

- We say a network is in a *stable clustering state* if for any agent A_i in the network, $|\overline{C_i}| = 0$.
- We say a group of agents D form a *closed cluster* if for any agent $A_i \in D$, $|\overline{C_i}| = 0$, and for any $A_j \in C_i$, $A_j \in D$. Also, for any pair of agent A_i and A_j, $a_i = a_j$.
- We say an agent has a *free connection* when the agent broke one old connection and gains the right to connect to a new neighbor.
- We call an agent a *free action agent* when it is possible for the agent to become either a majority coordinating neighbor or a majority non-coordinating neighbor by connecting to different new neighbors.
- We define *whipping* as the following process: every time that a free action agent looks for new neighbors and only connects to the agent who performs different action from it. In this way, the agent will keep flipping between the two possible actions. Because the agent's neighbors might have a high tolerance for the relationships between them and the agent, even when they perform different actions, the connection will not be broken immediately. Therefore, we allow the agent's neighbor go through the whipping process too. We let the agent's neighbors keep flipping between the two possible actions so that each time the agent and its neighbors use a different action. In this way, the reward for these connections will be constantly 0. After at most k turns, the agent will lose all of its neighbors and become isolated.

5.2 Theorems and proofs

We propose the following assumption:

Assumption *In a two-action coordination game, if the network has not reached a stable clustering state yet, then there will always be connections broken between agents playing different actions. And there always exists agents adopting different actions with free connections.*

Theorem 1 *If in a network all agents are majority coordinating neighbors, then it is possible that the network will reach stable clustering state in k turns.*

Proof for Theorem 1 If we only allow the agent to connect to the same action agents, then after k turns the agents will only keep a connection with the neighbors who adopt the same action as them. In this way, for any agent A_i we see $|\overline{C_i}| = 0$, and the network reaches the stable clustering state. □

Theorem 2 *If a network has at least one closed cluster, then it is possible for the network to reach a stable clustering state within $g(n)$ turns, where n is the number of agents in the network.*

Proof for Theorem 2 We prove this theorem by case analysis. First we check whether there exists any majority non-coordinating agent. If one doesn't exist, then by Theorem 1 we know it's possible for the network to reach stable clustering state in k turns. If there exists at least one majority non-coordinating agent, we select one and check whether this agent is a free action agent. If not, we pull free connections from non-coordinating agents to make the agent a free action agent. Then we let all of its free connections connect to a closed cluster when they try to link to new neighbors. If there is no closed cluster at the moment, we can change the current agent into a closed cluster through the whipping process. In this way, the original majority non-coordinating agent forms a closed cluster by itself. When the majority non-coordinating agent tries to connect to the closed cluster, and if the agent has more free connections than the agents in the closed network, then we can let the majority non-coordinating agent go through the similar process we describe above, except this time we will make the agent keep one free connection at last and use that connection to connect to the closed cluster. In this way, the agent becomes part of the closed cluster. We repeatedly run the above steps until there is no more majority non-coordinating agents left in the network. Then by Theorem 1, the network can reach a stable clustering state in k turns. □

Theorem 3 *Given a pure coordination game, placing no constraints on the initial choices of action by all agents, and assuming that all agents employ the HWR rule, then the following holds:*

- *For every $\varepsilon > 0$ there exists a bounded number M such that if the system runs for M iterations then the probability that a stable clustering state will be reached is greater than $1 - \varepsilon$.*
- *Once the stable clustering state is reached, it will never be left.*
- *If a stable clustering state is reached then all of the agents are guaranteed to receive an optimal payoff from all connections.*

Proof for Theorem 3 Similar to the proof for Theorem 2, we check whether there exists any majority non-coordinating agent first. If one does not exist, then by Theorem 1 we know that there is a probability $p = 1/f(n)$ the network will reach a stable clustering state in the next turn. If there is at least one majority non-coordinating agent, then we repeat the processes described above in order to make the agent become a closed cluster or part of an existing closed cluster. For each agent, there is a probability $p = 1/g(n)$ that this process can happen in $h(n)$ turns. We repeat the above process until there is no majority non-coordinating agent or the network reaches stable clustering state.

As a result, if the system runs for $M = x \times [n \times g(n) \times h(n) + f(n)]$ iterations then the probability that a stable clustering state will not be reached is at most e^{-x}. Taking $x > -\log(\varepsilon)$ yields the desired result.

For the second part of Theorem 3, because there only exist connections between agents who perform the same actions when the stable clustering state is reached, every connection will be optimal and no agent will try to break any connection. Therefore the stable clustering state will never be left.

For the third part of Theorem 3, since all connections are optimal, all agents are also guaranteed to receive an optimal payoff from all connections. □

6 Experiment

6.1 Experiment settings

All experiments are conducted in an environment that has the following properties:

- All trials run in a random network having 1000 to 10,000 agents.
- The number of connections in the network remains the same throughout the trial.
- Every time a connection is broken, both agents have a 50% chance to gain the right to connect to a new neighbor. But only one of them will eventually make a new neighbor. This restriction guarantees the number of connections remains the same.
- All agents have a limited memory size.
- All agents adopt the ideal learning rule. That is, the agent will always choose the action that the majority of its neighbors used in the last turn. If the same number of neighbors adopts different actions, then the agent will not change its current action. In other words, the agents obey the simple majority rule.
- The agents can only see local information. In other words, they do not know the payoff matrix and the identities of their neighbors.
- Our domain is the two-action Pure Coordination Game. It is a simple game in which agents receive a reward in the event they choose the same strategy and a penalty in the event they choose different strategies. Table 1 shows the payoff matrix for the Pure Coordination Game we use.

Notice that there are two equal Nash Equilibrium in the game: (Cooperate, Cooperate) and (Defect Defect). The optimal strategy depends on the strategy of an agent's neighbor. However, when this game is played with multiple neighbors, the optimal strategy is the strategy adopted by the majority of an agent's neighbors.

All code was written, compiled and ran in Ubuntu 9.10 with GCC 4.4.1. One test trial with 10,000 agents and an average neighborhood size of 100 takes approximately 90 seconds to run for 30 time steps on a HP HDX-16 laptop. The laptop has an Intel Core 2 Dual-Core 2.13 GHz CPU and 3 GB RAM.

Table 1 Payoff matrix for the pure coordination game

	Cooperate	Defect
Cooperate	1, 1	0, 0
Defect	0, 0	1, 1

Table 2 Parameters for Experiment 1 (Fig. 1)

Number of Runs	50
Timesteps	30
Network size	5000
Neighborhood size	10
Reward discount c	0.9
Threshold θ	0.9
Memory size	30

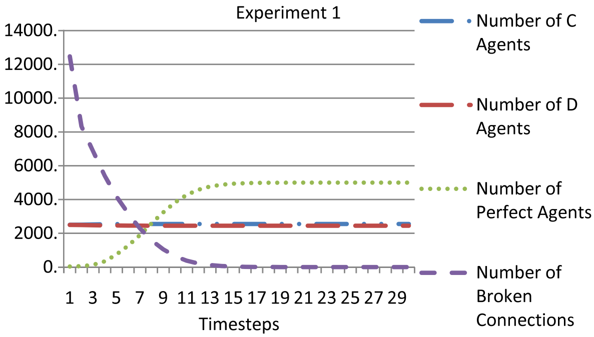

Fig. 1 Network converges into two clusters

6.2 Two-cluster stable clustering state

First, we show the emergence of the stable clustering state. Clusters refer to the groups of agents that have chosen the same action. Because the final outcome might have two different patterns with either one or two final stable clusters (i.e., C: Cooperate or D: Defect), we present the patterns in two parts. First, let's examine a situation in which the network eventually converges on a single cluster. The experiment is conducted using the parameters presented in Table 2.

Figure 1 shows the average results collected from 30 trials. The number of broken connections starts off at a little over 12,000 broken connections and the two clusters both linger at around the 2500 Agent mark. The number of broken connections and perfect agents intersect around time step 8. We see that at the end of each trial, all of the agents have split into two camps and entered into a unique stable clustering state. First, we notice that the lines that represent the numbers of C and D agents almost merge together in the middle of the plot. Because the pay-off matrix for C and D actions is symmetrical and there is no difference between these two actions except different symbolic names, the average numbers of C and D agents should be both around 2500, just as the plot shows. Here, in order to capture the dynamics of the network, we introduce a new term named *perfect agent*. An agent will be counted as a *perfect agent* only when all of its neighbors play the same action as it does. In Fig. 1, while the numbers of C and D agents remain almost constant, the number of perfect agents rises steadily from 0 to 5000 throughout the trial. The increasing number of perfect agents shows that the network is evolving from a random state towards a stable clustering state. At the same time, we see that the number of broken connections is steadily dropping towards 0. Because more and more agents become perfect agents, connections between neighbors are broken less and less frequently.

Table 3 Parameters for Experiment 2 (Fig. 2)

Number of Runs	50
Timesteps	30
Network size	5000
Neighbor size	100
Reward discount	0.9
Threshold	0.9
Memory size	30

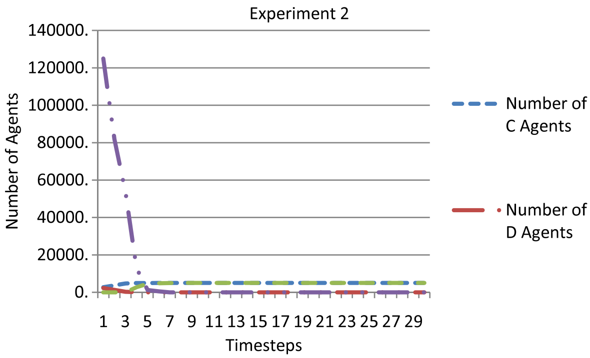

Fig. 2 Network converges into one cluster

6.3 One-cluster stable clustering state

Now we present the experiment results that all of the agents adopted the same action lasting the end and merged into one cluster. In order to show the results more clearly, we have run 50 trials and selected 23 trials in which C eventually becomes the dominant action. The experiment is carried out using the parameters presented in Table 3.

In Fig. 2, the number of C agents rises steadily while the number of D agents drops, until the entire network converges into a single cluster. The number of perfect agents and broken connections behaves in a manner similar to that in Fig. 1. The difference between this experiment and Experiment 1 is that the number of neighbors one agent can have has increased. Because each agent receives more input per time step due to an increased number of friends, agents will be more likely to change their action to that of those around them. Therefore, in this experiment, because we increased the amount of neighbors, the network shows a clustering around one decision.

6.4 Pattern possibilities with various network sizes

The above experiments demonstrate that the network sometimes converges into one single cluster and sometimes into two clusters. After a series of simulations, we found out that the size of neighborhood has a significant influence over the probability that a network will converge on one cluster or not. In order to show this, we created 16 networks with varying friend sizes (but always maintaining 1000 agents as the network size) and ran 50 trials with each network. The parameters presented in Table 4 were used.

Table 4 Parameters for Experiment 3 (Fig. 3)

Number of Runs	50
Timesteps	30
Network size	1000
Neighbor size	Varies
Reward discount	0.9
Threshold	0.9
Memory size	30

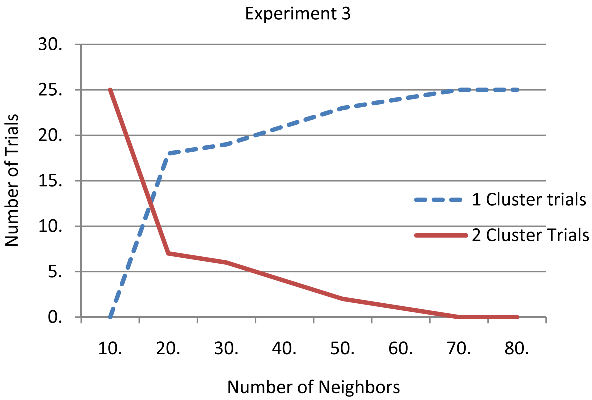

Fig. 3 Relationship between network size and network pattern

Figure 3 shows the number of one-cluster trials and two-cluster trials using each network. As the size of the network increases, the probability that the network will converge into one single cluster rises. There is a tipping point around neighborhood of size 15, when the neighborhood size is approximately 1.5% of the size of the whole network. In such cases, the network can converge on one or two clusters with approximately even possibility.

This phenomenon can be explained by the following. The size of a neighborhood directly determines the influence the majority action in the network places upon a single agent. The bigger the neighborhood is, the more the entire network influences the agent. Let's try to understand this from two extreme cases. Imagine the experiment is carried out in a complete network, in which every single agent is connected to all of the other agents in the network. In such a case, the agent's choice of action is affected by the actions of all the other agents, and, based on the simple majority rule, all the agents will immediately switch to the dominant action in this turn. On the other hand, if we let the agents keep a neighborhood of size 0, that means no agent has any neighbors, and then each agent will be under no influence from the network as a whole and will just stick to its current action. With similar logic, we see that the size of a neighborhood actually determines the influence each agent receives from the network and also the possibility that they will switch to the dominant action in the network. Therefore, the possibility that the agent will adopt the dominant action in the network and become one single cluster increases with increased neighborhood size. The following parameters in Table 5 were used for the second part of Experiment 4.

Table 5 Parameters for Experiment 4 (Fig. 4)

Number of Runs	30
Timesteps	30
Network size	2000
Neighbor size	10
Reward discount	0.9
Threshold	1.0
Memory size	15

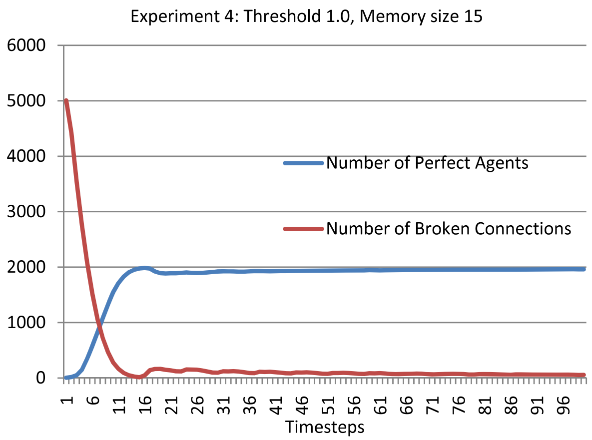

Fig. 4 Network with threshold 1.0 and memory size 15

6.5 Threshold and broken connection peak

As one might notice, in the preceding trials we used a threshold of 0.9. We chose a relatively low value here in order to make the agents more tolerant of their neighbors. (We discuss tolerance further in the later part of this paper.) In this case, as long as the weighted average reward from a certain neighbor is higher than 90% of the weighted average reward from all of the other agents, the neighbor will be counted as a good one and the connection will be reserved. In general, we expected that a higher threshold correlates with a lower tolerance. We have no concrete model for tolerance but we know that having higher thresholds makes agents both stricter about their and more likely to break up with their uncoordinated friends. This then implies that agents have a lower tolerance for their friends and exhibit no forgiveness despite having a good history with the said agent.

Figure 4 shows an interesting phenomenon that happens when the threshold is 1.0. The first time step shows the number of perfect agents and the number of broken connections, when the threshold is 1.0. In time step 16, the number of broken connections increases and the number of perfect agents decreases. The number of broken connections increases more until time step 19 and then starts to decrease with slight oscillations. However, up to time step 100, there are still broken connections in the network. The number of perfect agents decreases from time step 16 to 20 and starts to increase again with slight oscillations, but not all of the agents become perfect agents even by time step 100.

Table 6 Parameters for Experiment 4 (Fig. 5)

Number of Runs	30
Timesteps	30
Network size	2000
Neighbor size	10
Reward discount	0.9
Threshold	0.9
Memory size	15

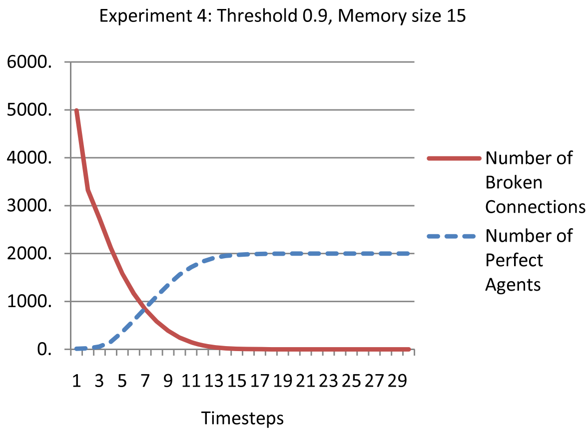

Fig. 5 Network with threshold 0.9 and memory size 15

At time step 16, because the memory size is only 15, agents begin to "forget" about the bad rewards they received in the first few turns. As a consequence, the weighted average reward for the agent rises quickly, and becomes higher than the weighted average reward for a certain few neighbors. Therefore, the agent chooses to break up with the neighbors, even though they have rewarded the agent in all of their interactions, except for the first few turns.

After time step 16, because of new connections, disconnections also happen. Agents acquire new friends though their actions could differ from the agent's. The agents should therefore disconnect the new relationship again. This is the reason that the network fails to have a network size of perfect agents even until time step 100.

By lowering the threshold, we can make the agents more tolerant of their neighbors and thus eliminate the peak. Also, we can change the memory size of the agents to control the time at which the peak appears. In order to support our argument, we ran a second trial with exactly the same setup, except at a threshold of 0.9. The parameters presented in Table 6 were used for the second part of experiment 4.

In Fig. 5, we see patterns similar to ones in Fig. 4, except without a peak. The number of broken connections and number of perfect agents intersect at time step 7 and diverge afterwards. The agents here only remember half of their entire interactions during one run of the simulation. With a threshold of 0.9, the agents are less strict and more tolerant of uncoordinated friends. Therefore, there is almost no connection being broken off once the network enters the stable clustering state. For these reasons, this particular experiment reaches a smooth equilibrium, unlike Fig. 4.

Table 7 Parameters for Experiment 4 (Fig. 6)

Number of Runs	30
Timesteps	30
Network size	2000
Neighbor size	10
Reward discount	0.9
Threshold	1.0
Memory size	20

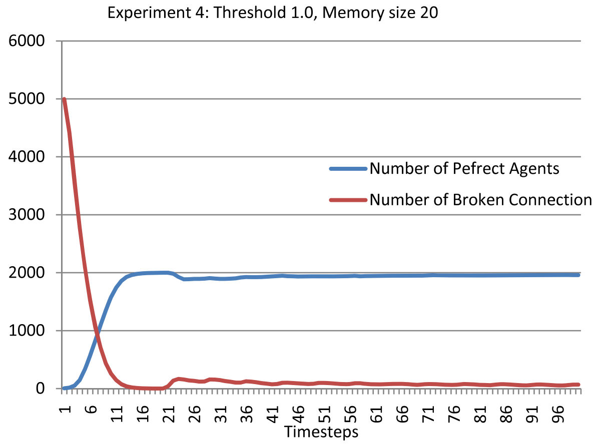

Fig. 6 Network with Threshold 1.0 and Memory size 20

Through this comparison, we can see that the existence of a peak (tolerance) is in direct relationship to the value of the threshold. We now can examine the relationship between the peak and the size of memory. The following trial is carried out with a memory size of 20 and a threshold of 1.0. The parameters presented in Table 7 were used for the third part of experiment 4.

As we can see in Fig. 6, when memory size grows from 15 to 20, the time for the peak to emerge is also pushed back by about 5 turns, thus further supporting our earlier stated theory.

6.6 Scalability

In this experiment, we examine the scalability of the network. We run different trials on various sizes of networks; in each network, the size of the neighborhood is 0.3% of the size of the whole network. The parameters presented in Table 8 were used for Experiment 5.

As we can see in Fig. 7, small networks generally converge faster than large networks. However, once a certain level is reached, the size matters little and the speed of convergence remains almost constant. Though the neighborhood size varies in each trial, the relative size to the network constantly remains 0.3%. It appears that the relative size of the neighborhood is the key factor that decides the convergence speed of the network. The following parameters in Table 9 were used for Experiment 6.

Table 8 Parameters for Experiment 5 (Fig. 7)

Number of Runs	30 for each network size
Timesteps	50
Network size	1000–10,000
Neighbor size	0.3% of network size
Reward discount	0.9
Threshold	0.9
Memory size	30

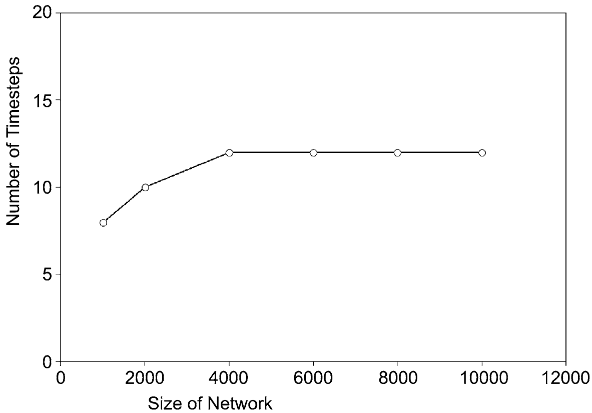

Fig. 7 Converging speed of 90% of network

Table 9 Parameters for Experiment 6 (Figs. 8 and 9)

Number of Runs	50
Timesteps	30
Network size	5000
Neighbor size	10
Reward discount	0.9
Threshold	0.9
Memory size	30

6.7 Comparison with static network

At the beginning of this paper, we claimed that we created this dynamic network in order to model modern dynamic networks that traditional static networks fail to model. In order to show this point more clearly, we carried out a comparison simulation between dynamic and static networks.

Figure 8 shows the results from a dynamic network and Fig. 9 shows the results forms a static network. First, the most obvious difference between these two trials may be the cumulative number of changed agents. The cumulative number of changed agents measures how many times the agents in the network change their actions. While the number for the dynamic network is only about 9000 changes, it is as high as 20,000 for the static network. Because in a dynamic network agents can maximize their rewards not only through changing actions but also through finding coordinating neighbors, agents need to change their actions less frequently than they do in static

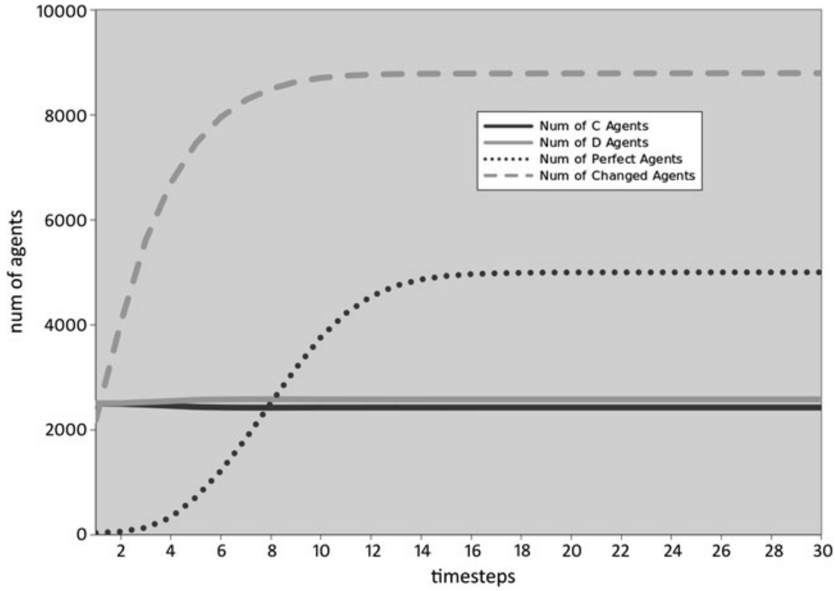

Fig. 8 Pattern in dynamic network

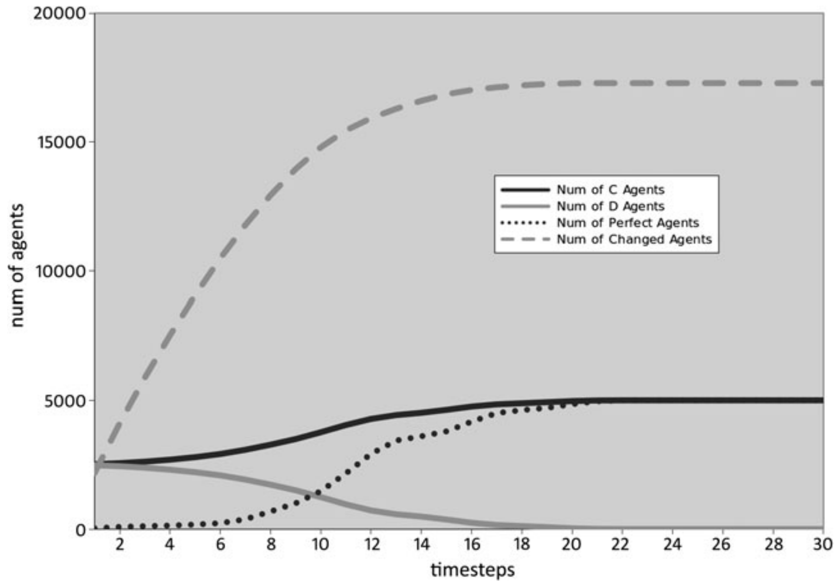

Fig. 9 Pattern in static network

network, in which all of the agents have to switch to one action in order to maximize the rewards received from the entire network. In addition, we noticed that a dynamic network also converges much faster than a static network. It only takes about 12

Table 10 Parameters for Experiment 7 (Fig. 10)

Number of Runs	30
Timesteps	30
Network size	2000
Friend size	20
Reward discount	0.95
Threshold	0.9
Memory size	15

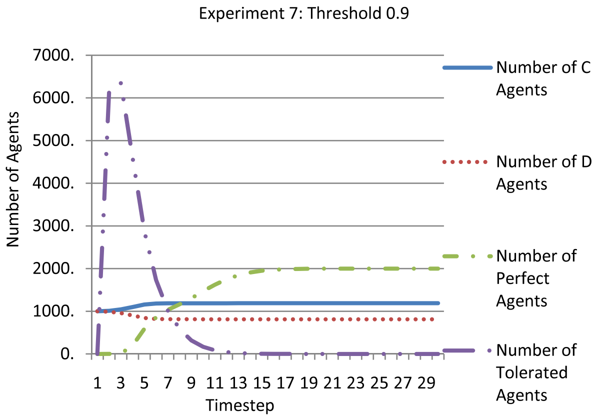

Fig. 10 Tolerance trial with threshold 0.9

rounds for a dynamic network to converge while it takes about 20 rounds for a static network. It appears that, by changing their neighbors, the agents can maximize their rewards in a much shorter period of time.

6.8 Tolerance

Because an agent evaluates its neighbors based on their cumulative rewards rather than the reward of one single time step, this experiment verifies the cumulative properties of the reward system for agent behavior. Occasionally we expect an agent to tolerate its friend's non-coordinating action because of their shared positive history. We also know that a higher threshold gives a lower tolerance. The parameters presented in Table 10 were used for the first part of experiment 7.

The parameters presented in Table 11 were used for the second part of experiment 7.

Figures 10 and 11 show two trials with thresholds of 0.9 and 0.7, respectively. As we can see from the plots, in both trials the agents are tolerating some of their neighbors. In addition, when the threshold is low (Threshold 0.7, Tolerated Agents around 9000 at peak), agents are tolerating more neighbors than they do with a high threshold (Threshold 0.9, Tolerated Agents around 6000 at peak).

Table 11 Parameters for Experiment 7 (Fig. 11)

Number of Runs	30
Timesteps	30
Network size	2000
Friend size	20
Reward discount	0.95
Threshold	0.7
Memory size	15

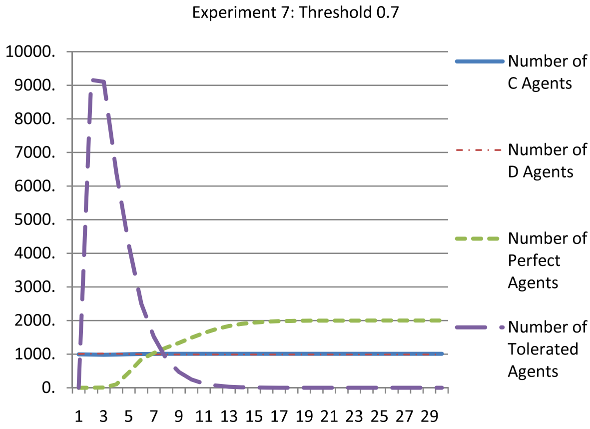

Fig. 11 Tolerance trial with a threshold of 0.7

7 Discussion

In Sect. 5, we proved that a stable clustering state emerges given a long enough period of time. However, as we can see from the experiments, the network usually converges very quickly. In most cases, the network can reach the stable clustering state within 20 time steps. This phenomenon is not uncommon, considering the nature of the HWR rule. Because an agent will not change its own action when there are an equal number of neighbors with different actions, the agent is more likely, statistically speaking, to stay with the current action during each time step. In other words, agents have a natural tendency to stick to one action. Once the agent in lodged in a coordination action dominant neighborhood, it becomes unlikely that the agent would change its action again. In order to make a change happen, a significant number of neighbors need to change actions, which is unlikely given the stable natural tendency of the agent. In addition, the more coordination neighbors an agent has, the more stable the agent is. Conversely, an action-switching state, such as the whipping state described in the proof, is very unstable. The longer the whipping process is, the less likely it is that the whole process will happen. Given the above reasoning, the fast convergence speed should be normal. This phenomenon further supports the strong correlation between the HWR rule and the observed pattern. In fact, such examples are not hard to find in everyday life. When a group of people goes out for lunch, each person in the party usually has a preferred place. Even still, given enough time to discuss and debate the matter, only one restaurant will remain. In order to dine altogether, people in the group will quickly abandon their personal preferences. Though our work is mainly theoretical, our assumptions are based on real-life scenarios. The HWR rule should be able to model any real-life scenario that meets our assumptions. In particular, we expect to see similar patterns emerge in real-life communities in which members behave in a manner comparable to a pure coordination game, choosing their actions based on the simple majority rule, and granting the right to choose their neighbors. For example, our model can be used to study the formation of interest groups, social activity clubs or even political parties, in which people tend to stay close to others who share the same interests or opinions. The parameters of the model can be further modified to fit different scenarios.

8 Conclusion

The HWR rule enables agents to increase their rewards not only by switching actions but also by updating neighborhoods. As our results show, agents adopting the HWR rule are able to change the network structure to maximize their own interests. Even though such agents behave selfishly, the network nonetheless reaches a Pareto-optimum social convention in the end.

Although the HWR rule is simple and easy to understand, we have observed a wide range of patterns emerging from our experimental results. Future research would do well to explore more fully the potential of this dynamic network model. This future research could pursue two paths.

The first avenue of exploration is the phenomenon of tolerance. As we have seen in the network, the agents adopting the HWR rule occasionally tolerate bad neighbors. In fact, this phenomenon was expected when we first designed the HRN and HWR rules. The point of allowing agents to evaluate their neighbors based on history rather than the most recent action is to allow the agents to keep the good neighbors who once were non-rewarding for very few turns. Though this phenomenon was observed in the network, it did not bring significant changes to the network structure. Moreover we had little control over its influence. The primary cause of this effect is the simple majority rule. Although the simple majority rule has been widely acknowledged and adopted in much research, the assumption it makes is rather unrealistic. Under the simple majority rule, agents should immediately switch their action once they have observed a majority performing a different action. However, this is apparently not the case in the real world. Human beings tend to be much more stubborn about their choices. People will not change their beliefs immediately after they see a majority of different beliefs. It usually takes a long time to change one's habits or beliefs, if it's possible at all. Therefore, in order to replicate the stubbornness of human beings, one could include in the model a new parameter called *resistance*. The resistance, r, would be an integer value larger than or equal to 1. It would simply refer to how many turns an agent could stay in a non-coordinating neighborhood until it changed its action. For example, if $r = 3$, then the agent would need to play with a majority of non-coordinate neighborhood three consecutive turns in order to change its action. We believe that by introducing the resistance value, tolerance could play a larger and much more important role in the model.

A second avenue of research would be to open up the constraints on our model. For instance, agents could be allowed in the current model to be added and removed, thus creating a new model dynamic from a different aspect. One could then compare the patterns of this new model with our existing model and study the differences. One could also allow the agents to play games other than the Pure Coordination Game, and study what social norms emerge with different games. Having done such work, we could further explore the possibilities of dynamic networks in various ways.

Acknowledgements This work was supported in part by the U.S. National Science Foundation under Grants IIS 0755405 and CNS 0821585. We would also like to thank Sophia Goreczky and KangChon Kim for their contributions to the experiment.

References

Acemoglu D, Ozdaglar A, ParandehGheibi A (2010) Spread of (mis)information in social networks. Games Econ Behav 70:194–227

Axelrod RM (1997) The complexity of cooperation: agent-based models of competition and collaboration. Princeton University Press, Princeton

Banks D (2009) Dynamic network models: introduction to new and interdisciplinary approaches. Comput Math Organ Theory 15:259–260

Borenstein E, Ruppin E (2003) Enhancing autonomous agents evolution with learning by imitation. AISB J 1(4):335–348

Centola D (2009) Failure in complex social networks. J Math Sociol 3(3):64–68

Chen Y, Bressler SL, Ding M (2009) Dynamics on networks: assessing functional connectivity with Granger causality. Comput Math Organ Theory 15:329–350

Davidsson P (2002) Agent-based social simulation: a computer science view. J Artif Soc Soc Simul 5(1)

Delgado J (2002) Emergence of social conventions in complex networks. Artif Intell 171–185

Dignum F (1999) Autonomous agents with norms. AI Law 1–15

Dorogovtsev SN, Goltsev AV, Mendes JFF (2008) Critical phenomena in complex networks. Rev Mod Phys 80:1275

Ehrhardt G, Marsili M, Vega-Redondo F (2008) Emergence and resilience of social networks: a general theoretical framework. Ann Econ Stat 86

Epstein J (1999) Agent-based computational models and generative social science. Complexity 4(5)

Erve I, Roth AE (1998) Predicting how people play games: reinforcement learning in experimental games with unique mixed equilibria. Am Econ Rev 848–881

Fararo TJ, Sunshine MH (1964) A study of a biased friendship net. Syracuse Univ. Press, Syracuse

Flache A, Macy MW (2011) Small worlds and culture polarization. J Math Sociol 35(1):146–176

Frantz TL, Cataldo M, Carley KM (2009) Robustness of centrality measures under uncertainty: examining the role of network topology. Comput Math Organ Theory 15:303–328

Handcock MS, Morris M (2009) A curved exponential family model for complex networks. Comput Math Organ Theory 15:294–302

Hoff PD (2009) Multiplicative latent factor models for description and prediction of social networks. Comput Math Organ Theory 15:261–272

Jiang Y, Toru I (2007) A model for collective strategy diffusion in agent social law evolution. In: International joint conference on artificial intelligence

Jin E, Girvan M, Newman MEJ (2001) The structure of growing social networks. Phys Rev

Jones JD (2008) Heterogeneity of degree and the emergence of cooperation in complex social networks. Emergence 10(4):46–54

Kempe D, Kleinberg J, Tardos E (2003) Maximizing the spread of influence in a social network. In: Proceedings of KDD 2003, Washington, DC

Kittok JE (1995) Emergent conventions and the structure of multi-agent systems. In: Nadel L, Stein D (eds) Lectures in complex systems. Addison-Wesley, Reading

Lewis DK (1969) Convention: a philosophical study. Harvard Univ. Press, Cambridge

Liljeros F, Edling CR, Amaral LAN, Stanely HE, Aberg Y (2001) The web of human sexual contacts. Nature 907–908

Lopez-Pintado D (2008) Diffusion in complex social networks. Games Econ Behav 62:573–590
Molchanov S, Whitmeyer JM (2011) Two Markoc models of the spread of rumors. J Math Sociol 34(3):157–166
Newmann MEJ (2003) The structure and function of complex networks. Society for Industrial and Applied Mathematics, Philadelphia
Pujol JM, Flache A, Delgado J, Sangüesa R (2005a) How can social networks ever become complex? Modeling the emergence of complex networks from local social exchanges. J Artif Soc Soc Simul 8(4)
Pujol Jm, Delgado J, Sangüesa R, Flache A (2005b) The role of clustering on the emergence of efficient social conventions. In: International Joint Conference on Artificial Intelligence, pp 965–970
Redner S (1998) How popular is your paper? An empirical study of the citation distribution. Eur Phys 131–134
Shoham Y, Tennenholtz M (1997) On the emergence of social conventions: Modeling, analysis and simulations. AI 139–166
Skyrms B (2004) The stag hunt and evolution of social structure. Cambridge University Press, Cambridge
Turner A, Penn A (2002) Encoding natural movement as an agent-based system: an investigation into human pedestrian behaviour in the built environment. Environ Plan B, Plan Des 29:473–490
Vance EA, Archie EA, Moss CJ (2009) Social networks in African elephants. Comput Math Organ Theory 15:303–328
Walker A, Wooldridge M (1995) Understanding the emergence of conventions in multi-agent systems. In: Lesser V (ed) Proc. 1st international conference on multi-agent systems. AAAI Press, Menlo Park, pp 384–389
Watts DJ (1999) Small worlds. Princeton University Press, Princeton
Younger S (2004) Reciprocity, normative reputation, and the development of mutual obligation in gift giving societies. J Artif Soc Soc Simul
Zimmermann M, Eguiluz V (2005) Cooperation, social networks and the emergence of leadership in a prisoners dilemma with adaptive local interactions. Phys Rev
Zhang Y, Leezer J (2009) Emergence of social norms in complex networks. In: Symposium on social computing applications (SCA09), The 2009 IEEE international conference on social computing (SocialCom-09), Vancouver, Canada, 29–31 August 2009, pp 549–555

Yu Zhang received her Ph.D. in Computer Science at Texas A&M University in 2005. She is currently Associate Professor in the Department of Computer Science at Trinity University and Director of the Laboratory for Distributed Intelligent Agent Systems. Dr. Zhang's research falls within Agent-Based Modeling and Simulation. She is Editor-in-Chief of The International Journal of Agent Technologies and Systems, and in the editorial board for several journals including The Transaction of Simulation, and in program committee for over 10 conferences and technical groups, such as Autonomous Agents and Multi-Agent Systems.

Yu Wu is a graduate student in the Department of Computer Science in Stanford University. He got his bachelor degree in computer science at Bard College. His research interests include Artificial Intelligence, Machine Learning and Social Simulations.