Evolution of Cooperation through Genetic Collective Learning and Imitation in Multiagent Societies

Honglin Bao[1,2], Qiqige Wuyun[1] and Wolfgang Banzhaf[1,2],

[1]Department of Computer Science and Engineering
[2]NSF Beacon Center for the Study of Evolution in Action
Michigan State University, East Lansing, MI 48824
{baohongl,wuyunqiq,banzhafw}@msu.edu

Abstract

How to facilitate the evolution of cooperation is a key question in multi-agent systems and game-theoretical situations. Individual reinforcement learners often fail to learn coordinated behavior. Using an evolutionary approach for selection can produce optimal behavior but may require significant computational efforts. Social imitation of behavior causes weak coordination in a society. Our goal in this paper is to improve the behavior of agents with reduced computational effort by combining evolutionary techniques, collective learning, and social imitation techniques. We designed a genetic algorithm based cooperation framework equipped with these techniques in order to solve particular coordination games in complex multi-agent networks. In this framework, offspring agents inherit more successful behavior selected from game-playing parent agents, and all agents in the network improve their performance through collective reinforcement learning and social imitation. Experiments are carried out to test the proposed framework and compare the performance with previous work. Experimental results show that the framework is more effective for the evolution of cooperation in complex multi-agent social systems than either evolutionary, reinforcement learning or imitation system on their own.

1. Introduction

A multiagent system (MAS) which consists of multiple interacting intelligent agents and their environment, is a computerized system for solving problems that are difficult or impossible for an individual agent to solve. Cooperation which has a long history in the application of game theory (Axelrod and Hamilton, 1981) assumes great importance in the field of multiagent system. In multiagent societies, cooperation represents an interaction among agents that can be evolutionarily advantageous to improve the performance of individual agents or the overall behavior of the society they belong to. Therefore, one of the main goals in multiagent societies is to achieve efficient cooperation among agents to jointly solve tasks or to maximize a utility function.

In order to realize such cooperation, some techniques developed in the field of machine learning have been introduced into various multiagent systems (Kapetanakis and Kudenko, 2002). Machine learning has been proven to be a popular approach to solve multiagent system problems because of the inherent complexity of these problems. Among machine learning techniques, reinforcement learning has gained much attention in the field of multiagent systems since it learns by trial-and-error interaction with its dynamic environment and can be used easily. However, several new challenges arise for reinforcement learning in multiagent systems. Foremost among these is that the performance of reinforcement learning is unsatisfactory in many real world applications. The learning algorithm may not converge to an optimal action combination. Some researchers showed that an adaptive strategy, called evolutionary reinforcement learning, which combines reinforcement learning with a genetic algorithm, could reach better performance than either strategy alone (Ackley and Littman, 1991). Some new forms of learning, e.g., observational, imitational, and communication-based learning (Taylor, et al. 2006, Savarimuthu, et al. 2011), also significantly promote information proliferation (Dittrich and Banzhaf, 2002) in more complex environments and can be used to solve complex distributed multiagent problems better than pure reinforcement learning approaches. Furthermore, ensemble methods are used to combine the advantages of multiple learning algorithms to obtain better performance than what could be obtained from any of them alone (Polikar, 2006). More recently, Yu et al. (Yu, et al. 2017) studied the role of reinforcement learning, collective decision making, social structure, and information diffusion in the process of the evolution of cooperation in the networked society.

Although previous work provided a strong basis to study the mechanisms behind the evolution of cooperation, existing work in this area has drawbacks. Individual reinforcement learners often fail to develop global coordinated behavior and can be trapped in local sub-optimal dilemmas. Using an evolutionary approach for strategy selection can produce optimal behavior but may require significant computational efforts. Behavior imitation always causes weak local coordination in a society, leading to local interactions between agents. This study is significantly different from other frameworks for the evolution of cooperation in previ-

ous studies, because of the hybrid policy of decision making of agents. Here we design a genetic algorithm based cooperation framework, which takes into account evolutionary selection, collective learning, and imitation, in order to solve some particular non-cooperative games in complex multiagent networks, overcome previous shortcomings, and produce an acceptable tradeoff in convergence rate and computation effort.

The final decision of an agent is influenced by three kinds of processes:

1) Evolutionary Selection (with inheritance and mutation): A population of agents plays a game with their neighbors (i.e., the agents which are directly connected with the focal agent) on the network for several iterations. The offspring generation will be reproduced from the parent generation according to the cumulative payoff distribution, and the most successful agents will pass on action to their offspring. Mutation will occur with a small probability during the inheritance process to create novelty.

2) Collective Learning: Agents on the network improve on their parents' actions and their original actions through a collective reinforcement learning algorithm with exploration and exploitation.

3) Imitation: Agents update the cumulative payoff, compare their cumulative payoff to neighbors, and adopt the actions of more successful agents as their own actions with a particular probability.

These three processes interact with each other, and can cause significant influence on the evolution of cooperation in the entire society.

The remainder of the paper is organized as follows. Section 2 introduces multiagent societies and the evolution of cooperation. Section 3 describes the proposed framework in multiagent societies. Section 4 presents experimental studies. Finally, Section 5 concludes the paper with some directions for future research.

2. Multiagent Societies and the Evolution of Cooperation

This section gives a description of multiagent societies and the evolution of cooperation.

Definition 1. A **Multiagent Society** can be represented as a networked undirected graph $G = (E, R)$, where $E = \{e_1, ..., e_n\}$ is a set of entities in the society (agents), and $R \subseteq E \times E$ represents a set of relationships, each of which connects two agents.

Definition 2. Given a multiagent society (E, R), the **Neighbors** of agent i, denoted as $N(i)$, are a set of agents so that $N(i) = \{e_j \mid \langle e_i, e_j \rangle \in R\}$ with $\langle e_i, e_j \rangle$ symbolizing a connection.

This paper adopts two typical topologies to represent a multiagent society, small-world networks and scale-free networks (Yu, et al. 2017). We use $SW_N^{k,\rho}$ to represent a small-world network, where k is the average size of the neighborhood of a node, ρ is the re-wiring probability to indicate the evolvability of small-world network, and N is the number of nodes. We use $SF_N^{k,\gamma}$ to represent a scale-free network, in which the probability that a node has k neighbors roughly equals to $k^{-\gamma}$. N is the number of nodes.

In this paper, we adopt the "Rules of the Road Game", a typical coordination game as an example to study the evolution of cooperation (Young, 1996). Consider two carriages meeting on a narrow road from opposite directions, having no context to decide on which side of the road to pass the other. If they choose differently, it will cause a head-on collision between them, and they receive a negative payoff. Only if they choose the same way, they can avoid a collision and receive positive payoff. To abstract from this realistic situation to virtual multiagent societies, agents are striving to establish a convention/law of coordinated action by choosing from an action space without any central controller. The payoff matrix is shown in Table 1.

Table 1: Payoff matrix of an n-action 2-player coordination game.

	Action 1	Action 2	...	Action n
Action 1	1,1	-1,-1	...	-1,-1
Action 2	-1,-1	1,1	...	-1,-1
...	...	...	...	...
Action n	-1,-1	-1,-1	...	1,1

There are multiple Nash-equilibria in this diagonal situation. Both of two players choose the same action, i.e., coordinated action. However, even purely rational players cannot choose the specific coordinated action without negotiation because they have no information to differentiate between strictly the same multiple equilibria. In realist, people can survive such social dilemma because there are laws or social norms for them to refer to. Our goal in this paper, is to train agents of a virtual society to choose the cooperative action without upper level steering and regulation.

3. The Proposed Framework

The overall proposed cooperation framework is shown in Algorithm 1. It constitutes a genetic algorithm (GA) based cooperation framework for MAS with collective decision making, learning and imitation to facilitate the evolution of cooperation used in some particular coordination games. This framework is set in a network structure such as a small-world network or a scale-free network. A population of agents plays the coordination game with their neighbors repeatedly and simultaneously in the network for several generations. Offspring generation i_o will be reproduced from parent generation i_p according to their cumulative payoff E_i distribution. The most successful agents pass on behavior to their offspring i_o, and mutation will change this behavior with a small probability η during inheritance, described in Subsection 3.1. The society information regarding

Algorithm 1: The proposed cooperation framework

```
Initialize multiagent network and parameters;
for each step t (t=1,...,T) do
    for each agent i (i=1,...,n_0) do
        for each neighbor j ∈ N(i) of agent i do
            Agent i plays the game with neighbor agent
            j and receives corresponding payoff r_i^j;
        end
        Agent i calculates the cumulative payoff E_i;
        Offspring generation i_o will be reproduced from
        parent generation i_p according to E_i;
    end
    for each parent agent i_p (i_p =1,...,n_0) do
        Parent i_p passes on behavior to the offspring i_o;
        Mutation will change it with a small probability
        η during inheritance;
    end
    The society information regarding nodes and edges
    will be updated;
    for each agent i in a new network do
        for each neighbor j ∈ N(i) of agent i do
            Agent i improves the behavior with a
            collective learning method with exploration
            and exploitation regarding neighbor j;
            Agent i and neighbor j update the
            cumulative payoff E_i' and E_j';
            Agent i imitates the action of neighbor
            agent j with a probability W;
        end
    end
end
```

nodes and edges will be updated regularly. Then agents will improve their actions (including inherited action and original action) through a collective reinforcement learning algorithm with exploration and exploitation, described in Subsection 3.2. This will often cause later generations to converge to optimal behavior in the coordination game (McGlohon and Sen, 2005). After collective reinforcement learning, there is an imitation phase. Agents update and compare their cumulative payoffs with neighbors, and imitate their neighbors' actions with a probability W, more detail in Subsection 3.3.

3.1. Selection, Inheritance and Mutation

This subsection describes the process of payoff-distribution based reproduction (i.e., selection), inheritance, and mutation.

Definition 3. Given a multiagent society (E, R), the **Action Space** of this society, denoted as N_a, is a set of actions available to choose from for all agents, so that $N_a = \{a_0, a_1, ..., a_\tau\}$. τ is the number of available actions.

In Algorithm 1, we first initialize the multiagent network and parameters. Each agent will take an action from action space N_a chosen randomly. Agent i plays the game with neighbor agent j repeatedly and receives a corresponding payoff r_i^j according to Table 1. Agent i calculates their cumulative payoff E_i. When agents are chosen to reproduce, their fitness is based on the relative cumulative payoff distribution P_i shown in Equation 1 (McGlohon and Sen, 2005).

$$P_i = E(i) / \sum_{j=1}^{n_0} E(j) \tag{1}$$

The probability θ_i of agent i being chosen to reproduce (i.e., fitness function) is shown in Equation 2.

$$\theta_i = \begin{cases} \mathrm{P}_i & \text{if } E(i) \geq 0 \wedge \sum_{j=1}^{n_0} E(j) > 0, \\ 1/\mathrm{n}_0 - P_i & \text{if } E(i) > 0 \wedge \sum_{j=1}^{n_0} E(j) < 0, \\ 0 & \text{if } E(i) < 0 \wedge \sum_{j=1}^{n_0} E(j) > 0. \end{cases} \tag{2}$$

The situation for $E(i) < 0 \wedge \sum_{j=1}^{n_0} E(j) < 0$ is complex. We set $|E_i|$ as the absolute value of E_i. For $E(i) < 0 \wedge \sum_{j=1}^{n_0} E(j) < 0$, the probability θ_i of agent i being chosen to reproduce is given in Equation 3.

$$\theta_i = \begin{cases} \mathrm{P}_i & \text{if } |E(i)| < |\sum_{j=1}^{n_0} E(j)|, \\ 0 & \text{if } |E(i)| > |\sum_{j=1}^{n_0} E(j)|. \end{cases} \tag{3}$$

Equation 2 and 3 are inspired by win-stay, lose-shift, a simple but insightful social strategy (Nowak and Sigmund, 1993). Here winning means a positive payoff, and loosing means a negative payoff. Winning individuals in a global losing environment should be given more chance to reproduce. Ordinary individuals just reproduce the ordinary number of offspring. Furthermore, loosing individuals should be punished in a positive society. We use fitness proportionate selection. Notice that there is no crossover or recombination in our model. Offspring i_o will be reproduced from parents i_p according to the fitness function. Notice:

1) If the cumulative payoff of the entire population is 0, i.e., $\sum_{j=1}^{n_0} E(j) = 0$, we will reinitialize the experiment;

2) If $\theta_i > 1$, we set $\theta_i = 1$.

After reproducing offspring based on fitness, parents simply pass on their behaviors to offspring. In this process, mutation will change the behavior of offspring with a small probability η. In this case a random behavior will be chosen rather than the inherited behavior. We set $\eta = 1\%$ (McGlohon and Sen, 2005).

3.2. Collective Learning

As shown in Algorithm 2, collective learning is proposed to improve the behavior (both inherited and original) in an extending network. All agents in the society interact repeatedly and simultaneously with their neighbors. In each time step, an agent uses a reinforcement learning algorithm to choose a best-response action for each neighbor. The best-response actions for all neighbors are then aggregated into

an overall action using collective voting methods, which will be described in details in 3.2.1. Local and global exploration and exploitation will be discussed in 3.2.2. The agent then plays the overall action with all of its neighbors and receives a corresponding payoff according to Table 1. The learning information for each neighbor is updated by the overall action and the corresponding payoff. The entire process of this algorithmic framework is shown in Figure 1. Here we just focus on the neighbors of agent i.

Algorithm 2: The collective learning framework

```
for each step t (t=1,...,T) do
    for each agent i (i=1,...,n) do
        for each neighbor j ∈ N(i) of agent i do
            Agent i has a Q function for each of its
            neighbours j;
            Agent i chooses a best-response action
            a_{i→j} regarding neighbor j using a
            Q-learning algorithm;
            //Local exploration;
        end
        Agent i aggregates all the actions a_{i→j} into an
        overall action a_i using ensemble learning
        methods;
        //Global exploration;
    end
    for each agent i (i=1,...,n) do
        Agent i plays action a_i with its neighbors and
        receives corresponding payoff r_i^{j'} for each
        interaction;
        Agent i updates learning information towards
        each neighbor using action-payoff pair (a_i, r_i^{j'});
    end
end
```

3.2.1. Collective Decision Making

After reproduction, in this new extending society, all agents first interact with their neighbors. We adopt a widely used reinforcement learning algorithm, Q-learning, to model this interaction. Its one-step updating rule is given by Equation 4. Here $\alpha \in (0, 1]$ is a learning rate, and $\lambda \in [0, 1)$ is a discount factor.

$$Q(s,a) \leftarrow Q(s,a) + \alpha[R(s,a) + \lambda \max_{a'} Q(s',a') - Q(s,a)] \tag{4}$$

As shown in Equation 4, an agent has a set of states and a set of actions. An agent performs an action a, transitions from state s to another new state s' and receives immediate reward $R(s,a)$. $Q(s,a)$ is the expected reward of choosing action a in state s at time step t. During the interaction, agents want to maximize the expected discounted reward $Q(s',a')$ to make decisions in the new state s' at time step $t+1$. The Q-function is learned during an agent's lifetime inherited to choose a best-response action based on the Q-value regularly.

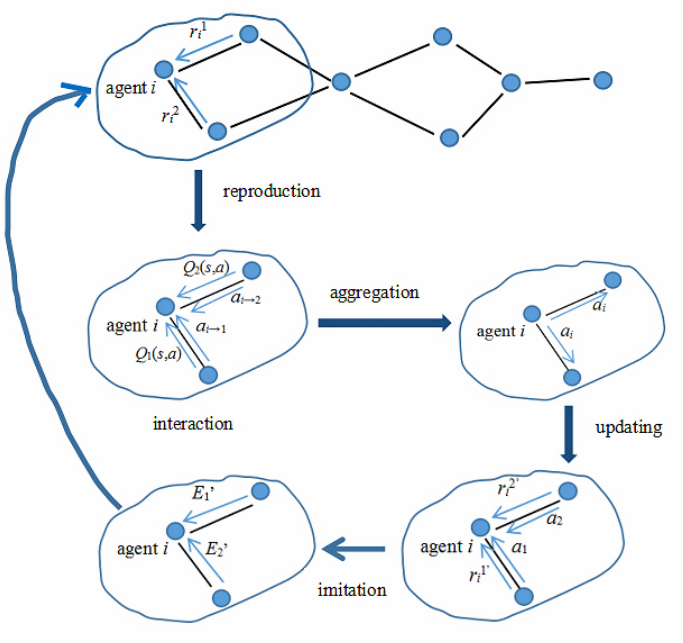

Figure 1: The entire process of our proposed framework. Agent i first plays the game and receives payoff r_i^1 and r_i^2 from two neighbors, respectively. After reproduction, agent i interacts with new neighbors, and chooses the best response action-reward pair $\{a_{i\to1}, Q_1(s,a)\}$ and $\{a_{i\to2}, Q_2(s,a)\}$. Then agent i aggregates $a_{i\to1}$ and $a_{i\to2}$ into an overall action a_i. Agent i keeps action a_i to play with neighbors and receives payoff $r_i^{1\prime}$ and $r_i^{2\prime}$. The cumulative payoff E_i, E_1', and E_2' of agent i and neighbors is updated. Agent i imitates neighbors according to the new cumulative payoff.

Each agent needs to aggregate all the best-response actions regarding its neighbors into an overall action. This is inspired by the opinion aggregation process in that people usually have seek for the suggestions from many other people before making a final decision. The opinion aggregation process can be realized by an ensemble learning method which combines multiple single-learning algorithms to obtain better performance than what could be obtained from any of them alone (Polikar, 2006).

The foremost method of collective voting is inspired by a simple political principle, **majority rule**. Consider that in a simple society (e.g., a undirected simple graph which represents the multiagent network we adopt in this paper), human beings are more keen to decide as the majority of their neighbors. So in this paper, when agents make final decisions, they consider the action which quantitatively dominates in the best-response action pool. More complex and realistic methods to make a final decision consider the weight of each neighbors, such as **performance-based weighted voting method** and **structure-based weighted voting method**.

For structure-based weighted voting, the weight of each neighbor is related to the degree of each neighbor. The focal agent will give higher weight to a neighbor with more connections. For performance-based weighted voting, the focal agents will consider previous interaction experience and will give higher weight to neighbors they trust. More detailed description of these collective voting methods can be found in (Yu, et al. 2017). In this study, we adopt majority voting as the opinion aggregation method.

3.2.2. Exploration and Exploitation

For pure greedy-learning, agents can be trapped easily in local sub-optima, and thus fail to learn the optimal behavior. During learning, an agent needs to strike a balance between exploitation of learnt knowledge and the exploration of unexplored environments in order to try more actions, escape from local sub-optima, and learn optimal behavior. In this paper, we propose **Simulated-annealing Exploration** for dealing with exploitation and exploration during learning.

Simulated Annealing (SA) is a non-linear technique for approximating the global optimum of a given function. We adopt an SA and combine it with traditional exploration. One step of SA exploration is given by Equation 5.

$$\mu_t = \mu_0 / \lg(1+t) \tag{5}$$

In Equation 5, μ_t is the exploration rate in the t^{th} round of simulation, and μ_0 is the initial exploration rate. At the beginning (t is small), exploration should be given higher weight to explore the unknown environment. As the algorithm continues (t increases), the probability of exploitation (i.e., $1-\mu_t$) increases determining that the agent will focus more on exploitation of learnt knowledge.

In Algorithm 2, during the interaction with neighbors, agents need to find a best-response action regarding each neighbor with a Q-learning method. At each time step t, regarding each neighbor j, agent i chooses the best-response action with the highest Q-value with a probability of $1-\mu_t$ (i.e., exploitation), or chooses an action randomly with a probability of μ_t (i.e., exploration). This occurs in the process of local interaction with neighbors. We call this process **Local SA Exploration**. When agents use specific ensemble methods to aggregate all the best-response actions into an overall action, agents choose the overall action under ensemble methods with a probability of $1-\mu_t$ (i.e., exploitation), or choose an action randomly with a probability of μ_t (i.e., exploration). This occurs in the process of overall aggregation. We call this process **Global SA Exploration**. A small average exploration rate (such as 10%) is kept throughout to conserve a small probability to explore.

3.3. Social Learning and Imitation

Social learning theory is connected with social behavior and learning and proposes that new behavior can be obtained by observing and imitating others' behavior (Bandura and Walters, 1977). In real life, people not only can learn through their individual trial-and-error experiences (i.e., individual Q-learning to determine best-response actions), but also seek suggestions or advice from others in a society (as mentioned in opinion aggregation in 3.2.1). Furthermore, they can also learn from the information directly provided by others through communication, observation, and imitation (Polikar, 2006).

We are inspired by social learning theory to add an imitation process after learning to promote the evolution of cooperation. After reproduction and learning, there is a new population with better performance in multiagent societies. In every time step, when agent i updates the cumulative payoff $E_i^{'}$, agent i in this new population adopts neighbor agent j's behavior, replacing its heritable behavior, with a probability W. Following Szabó and Tőke (Szabó and Tőke, 1998), we set:

$$W = \frac{1}{1+e^{-(E_j^{'}-E_i^{'})/K}} \tag{6}$$

Here, $E_i^{'}$ and $E_j^{'}$ are the cumulative payoff of agent i and neighbor j after updating. K represents some noise which is introduced to consider irrational choices. For $K=0$ agent i adopts neighbor j's strategy if $E_j^{'} > E_i^{'}$. Here we set $K=0.1$.

4. Experimental Studies

The purpose of this experiment is to study the evolution of cooperation in the proposed framework. The performance standards are the asymptotic percentage of cooperative actions (i.e., how many agents in the society can reach a final consensus, e.g., choose a specific action as coordinated action from action space) and convergence time (i.e., the time needed to reach such a consensus). We want to produce an acceptable trade off in both of them.

4.1. Experimental Settings

We use the Watts-Strogatz model (Watts and Strogatz, 1998) to generate a small-world network, and use the Barabasi-Albert model (Albert and Barabasi, 2002) to generate a scale-free network. In order to use the Barabasi-Albert model, we start with 2 agents and add a new agent with 1 edge to the network at every time step. Because of the re-wiring probability ρ, this approach generates a scale-free network following a power law distribution with an exponent $\gamma = 3$. We set the maximum number of edges to 1,000,000 for network evolution. Mutation rate η in inheritance is 0.01. Individual Q-learning rate α is 0.1. Average exploration rate in SA exploration is 0.1. The initialized SA exploration rate μ_0 is 0.144. Noise in imitation is set to 0.1. In this study, unless stated otherwise, we use the small-world network as the default network topology because it

can evolve into many kinds of networks, and local SA exploration as the exploration mode. All results are averaged over 100 independent time step.

4.2. Results and Analysis

Influence of action spaces Here, we vary action space in the set $N_a = \{2, 10, 20\}$ in network $SW_{100}^{12,0.8}$ to study its influence on the evolution of cooperation. According to Table 1, only when two agents choose the same action they will receive a payoff of 1. Otherwise, they receive a payoff of -1. Results in Figure 2 show that a larger number of available actions causes a delayed convergence of coordinated action. This is the result of learning and imitation regarding neighbors. Because of a larger number of actions, agents need more local interactions to learn an optimal behavior regarding neighbors and choose the best behavior among this large action pool to imitate neighbors. It may produce more varied local distributed sub-coordination which emerges from varied local interaction among agents and their neighbors, leading to diversity across the society. It thus takes a longer time for agents to overcome this diversity and achieve a final coordination, and thus the evolution of cooperation is prolonged in the entire society.

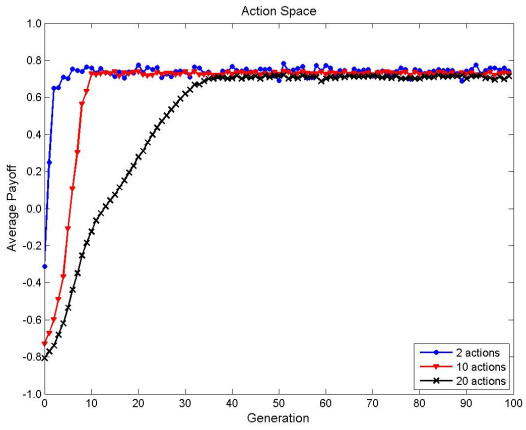

Figure 2: Influence of number of actions.

The Influence of Single Mechanism Broadly, given that four very different mechanisms, i.e., genetic algorithm (GA), reinforcement learning (RL), collective decision making (CDM), and imitation, are being used, we want to give some forms of direct comparison of what each mechanism contributes to the dynamics and convergence properties in order to understand the role that each mechanism plays in this system and how they interact.

We fix the action space to $N_a = 10$. The influence on the evolution of cooperation under different mechanism combinations is shown in Figure 3. Without GA situation means that only collective learning and imitation occur in a fixed,

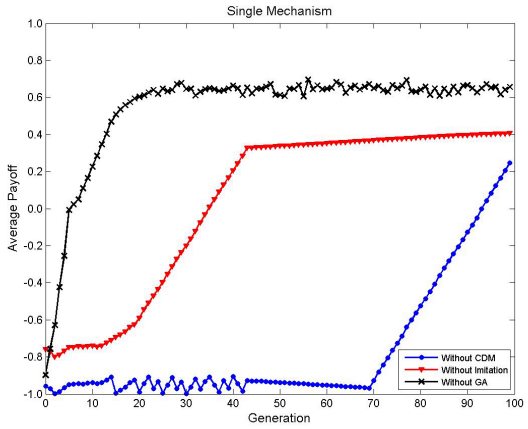

Figure 3: Evolutionary dynamics under different combinations of mechanism.

static agent society; without CDM situation means that after choosing the best-response actions from neighbors, the focal agent simply determines one action randomly as the overall action; without imitation situation means that only evolutionary selection and collective learning occur in an extending agent society. From Figure 3, we can draw these conclusions:

1) Collective decision making (opinion aggregation) and imitation will significantly facilitate the evolution of cooperation, especially collective decision making.

2) Evolutionary selection does cause influence both on the convergence speed and convergence rate, but not as dramatic as collective decision making or imitation.

Notice that in Figure 3, we do not show the evolutionary dynamics in this system without reinforcement learning (RL), i.e., the focus agent simply aggregates the original action or inherited action of their neighbors into an overall action without any RL-based improving. Since we could not get any convergence curves in 100 generations during experiments. We can say:

3) In this system, reinforcement learning to make better decisions is the most important step to promote the evolution of cooperation. It is dominated by one of these four modalities and contributes most to the rate at which cooperation emerges.

Comparison of mutation and two types of exploration As shown in Subsection 3.1 and 3.2.2, we should test the single influence of mutation, local SA exploration, and global SA exploration and compare them.

We test the situation under 4-action space, i.e., action 0,..., action 3 respectively. Figure 4 shows the asymptotic percentage of cooperative actions (action 0) adopted by the agents when cooperation evolves in the entire society. Initially, each agent randomly chooses an action from action space, so there are about 25% of all agents to choose each

action respectively. As our framework moves on, the number of agents who choose action 0 as the cooperative action finally reaches more than 90% in the situation with SA exploration (both local and global). This result means that more and more agents have reached a consensus on that action 0 should be the cooperative action. From Figure 4, we can see that the fraction of cooperators in the society using collective learning with local SA exploration mode is almost 100% which means that almost all the agents have reached a consensus on which action should be the cooperative action. The framework works in the entire society.

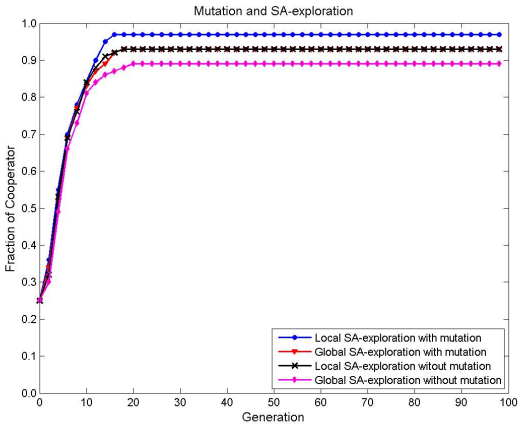

Figure 4: Fraction of cooperators under different exploration and mutation methods.

We further study Figure 4 and we can draw these conclusions:

1) Local exploration is better than global exploration.

The fraction of cooperators using collective learning with the global exploration mode is much lower than that using collective learning with the local exploration mode. This is because agents explore the environment with a probability of 0.1. However, as agents using local exploration to explore the environment locally (i.e., choosing irrational action during local interaction) and aggregate to an overall action collectively, the randomness caused by the exploration can be removed. In global exploration, agents explore globally when they aggregate all best-response actions into an overall action, the randomness will be kept.

2) Mutation is necessary.

The fraction of cooperators with mutation is higher than that without mutation. Although sometimes mutation has a bad influence, indeed, it is the source of novelty.

For both exploration and mutation, it seems notable that removing mutations and switching between local and global exploration does not seem to change the rate at which a consensus action is discovered (i.e., the transient part of the curve), but only shows up in the different asymptotic percentage of cooperators. It indicates these factors are helping the system avoid getting stuck in a local optimum near the completely converged state but can not show their roles on promoting cooperation clearly. The reason we guess is that, in this system, collective decision making causes the dramatic influence on convergence speed, as shown in Figure 3. So the weak influence of mutation and exploration on the convergence speed can not be found very dramatically.

Comparison with Previous Work We mainly compare the performance of our model with (Yu, et al. 2017). As shown in Figure 5, we follow the previous parameter settings ($N_a = 10$), our framework has better performance than previous study. We additionally test other situations with different action space, the results show the same trends. It indicates that our model works for the evolution of cooperation in the entire society. It is indeed effective for the robust evolution through combining evolutionary selection, individual learning, collective voting, and social imitation.

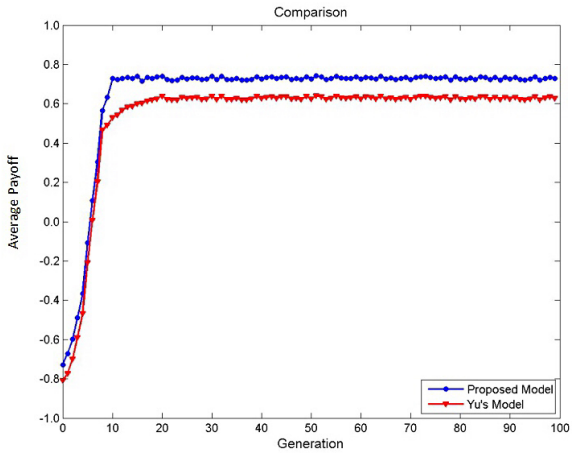

Figure 5: Comparison with (Yu, et al. 2017). Yu's work is mainly based on collective reinforcement learning and information diffusion (i.e., communication-based social learning, agents sharing Q table to communicate).

Through our experimental analysis, we also find that there is not much difference in the efficiency of the evolution of cooperation in different sizes of agent population, different opinion aggregation methods, and different network structures. To summarize, for robust cooperation evolving in networked agent systems, the potential key factors are:

1) **the way how agents interact with each other**. This is also called interaction protocol. For instance, interacting randomly in a population or interacting with neighbors in a network; what game-theoretical situations the interaction is based on (as shown in the payoff matrix in Section 2).

2) **the way how agents update their learning information through interaction**, i.e., what learning strategies (e.g., collective Q-learning, WoLF-PHC, and fictitious play) do agents use to update their learning information?

3) **the way how agents diffuse their learnt information**, e.g., communication-based social learning, imitation-based

social learning, and observation-based social learning.

4) **whether the entire population evolves in a better direction**. Evolving to improve the entire fitness (e.g., reproducing offspring with better performance to increase the entire average fitness) represents an enhancement in the evolution of cooperation.

5. Conclusion and Future Work

Evolution of cooperation has been extensively studied in MASs. Existing work in this area, however, has some drawbacks especially considering that evolutionary selection, reinforcement learning, collective decision making, and behavior imitation have dramatic influences on the evolutionary process. This paper proposes a genetic algorithm based framework with collective learning and imitation in multiagent networks. The goal of this work was to investigate whether cooperation can be facilitated by these factors, and whether our framework has a better performance than previous studies. In this paper, we want to make an acceptable tradeoff in both convergence speed and convergence rate. Although other papers report a convergence to 100% of cooperative actions in similar systems even when just a single learning method, the extreme computing resource, e.g. long term evolutionary generations, is not acceptable. Experiments were carried out to test the proposed framework in different parameter settings and environments. Experimental results show that our mechanism is indeed effective for the evolution of cooperation in multiagent networks and that our framework has better performance (both convergence speed and rate) than previous work.

This paper is just an initial step for this research. The long term goal is to design some robust mechanisms for efficiently coordinated control of more realistic large scale distributed system. To realize this goal, much work still needs to be done. For example, the time-varying relationships between agents, e.g., supervisor and subordinate, and adaptive interactions, e.g., disconnecting punishment mechanism, can be added into existing network to generate dynamical hierarchical multiagent societies.

Acknowledgments

This paper is based on H. B.'s previous work with the members of Interests Group on Multi Agent Systems (IGMAS) at Dalian University of Technology, Dalian, China. H. B. thanks Dr. Chao Yu and Mr. Hongtao Lv for help in building a solid foundation in the field of multiagent systems. The authors also acknowledge the support from High Performance Computing Cluster (HPCC) of Michigan State University.

References

Ackley, D., and Littman, M. (1991). Interactions between learning and evolution. *Artificial Life II*, 10, 487-509.

Albert, R., and Barabasi, A. L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1), 47.

Axelrod, R., and Hamilton, W. D. (1981). The evolution of cooperation. *Science*, 211(4489), 1390-1396.

Bandura, A., and Walters, R. H. (1977). Social learning theory.

Dittrich, P., Kron, T., and Banzhaf, W. (2002). On the scalability of social order-modeling the problem of double and multi contingency following Luhmann. *Journal of Artificial Societies and Social Simulation*, 6(1).

Kapetanakis, S., and Kudenko, D. (2002). Reinforcement learning of coordination in cooperative multi-agent systems. *AAAI/IAAI*, 2002, 326-331.

McGlohon, M., and Sen, S. (2005). Learning to cooperate in multi-agent systems by combining Q-learning and evolutionary strategy. *International Journal on Lateral Computing*, 1(2), 58-64.

Nowak, M., and Sigmund, K. (1993). A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner's Dilemma game. *Nature*, 364(6432), 56.

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, 6(3), 21-45.

Savarimuthu, B. T. R., Arulanandam, R., and Purvis, M. (2011). Aspects of active norm learning and the effect of lying on norm emergence in agent societies. In *International Conference on Principles and Practice of Multi-Agent Systems* (pp. 36-50). Springer, Berlin, Heidelberg.

Szabó, G., and Tőke, C. (1998). Evolutionary prisoner's dilemma game on a square lattice. *Physical Review E*, 58(1), 69.

Taylor, M. E., Whiteson, S., and Stone, P. (2006). Comparing evolutionary and temporal difference methods in a reinforcement learning domain. *In Proceedings of the 8th annual conference on Genetic and Evolutionary Computation* (pp. 1321-1328). ACM.

Watts, D. J., and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393(6684), 440.

Young, H. P. (1996). The economics of convention. *Journal of Economic Perspectives*, 10(2), 105-122.

Yu, C., Wang, Z., Lv, H., Bao, H., and Li, Y. (2017). Collective Learning and Information Diffusion for Efficient Emergence of Social Norms. In *Multi-agent and Complex Systems* (pp. 193-210). Springer, Singapore.