Topology and memory effect on convention emergence

Daniel Villatoro*, Sandip Sen† and Jordi Sabater-Mir*
*Artificial Intelligence Research Institute (IIIA)
Spanish National Research Council (CSIC)
Bellatera, Barcelona, Spain
†Department of Mathematical and Computer Science
University of Tulsa
Tulsa, Oklahoma, USA

***Abstract*—Social conventions are useful self-sustaining protocols for groups to coordinate behavior without a centralized entity enforcing coordination. We perform an in-depth study of different network structures, to compare and evaluate the effects of different network topologies on the success and rate of emergence of social conventions. While others have investigated memory for learning algorithms, the effects of memory or history of past activities on the reward received by interacting agents have not been adequately investigated. We propose a reward metric that takes into consideration the past action choices of the interacting agents. The research question to be answered is what effect does the history based reward function and the learning approach have on convergence time to conventions in different topologies. We experimentally investigate the effects of history size, agent population size and neighborhood size or the emergence of social conventions.**

I. Introduction

Social norms such as driving on the left side of the road or not stepping in front of other people in line are prevalent in human groups and societies. Such norms are conflict resolution strategies that develop from the population interactions instead of a centralized entity dictating agent protocol. History of interaction is then instrumental for norm evolution. Learning algorithms incorporate history of interaction into their calculations, but reward metrics are typically static and independent of the agent histories. Norm evolution is dependent upon the exertion of social pressure by the group on aberrant individuals. It is through learning via repeated interactions that social pressure is applied to individuals in the group. However, a reward metric based on the current interaction does not necessarily model the full context or capture the persistent nature of social pressure in human societies. In particular, society often uses past history to judge individuals and hence actions have future consequences in addition to immediate effects. Accordingly, we propose a reward structure based upon the agent's interaction history as a more appropriate alternative to the single interaction reward metric normally used. In our model agents are rewarded based upon the conformity of action between two agents, such that the agent who has the most of the majority interaction receives higher reward. Hence, both interacting agents' history of actions are used to calculate each individual's payoff from an interaction. We investigate how this history, and in particular, its size (memory size) affects the emergence of social conventions in different types of societal structures.

We are also keenly interested in understanding how agent relationships and social connections affect the success and rate of adoption of social norms. We represent different societal connection topologies by different network types in which the network links represent interactions between agents. Given a connection topology, agents repeatedly play a two-player game with the reward for interaction based on their respective action histories. We believe that the underlying topology of the society is a key factor in determining the convention emergence process. In this work we will experiment on different types of topologies in order to observe, compare and analyze their effects and dynamics of reaching social conventions.

The structure of this article is as follows: we review the previous and related work in the area of emergence of social conventions in multiagent systems in Section II; in Section III we present the agent interaction and reward model that we have used; experimental results are presented in Section IV; conclusions from the analysis of the results are presented in Section V, and finally we present the future work in Section VI.

II. Previous Work

Sen and Airiau [1], [2] explored norm emergence where interaction rewards were not dependent on previous interactions. That work is focused on the problem of coordination of two cars arriving at an intersection. Each agent can choose to "go" or "yield" to the other agent. The reward metric is designed so that if each agent chooses the same action, they receive small payoff but if agents choose opposite actions, they receive a large payoff. So if the row and the column agents both "go" they both receive a poor payoff, but only one player choosing to go will yield a relatively high payoff for both. Depending on which player chooses to yield, two possible effective social norms can be established. Each agent in an interaction was randomly chosen from the population. Agents learned to adopt a consistent social norm from repeated interactions with other agents in the population. The history of interaction does not directly affect the reward agents receive.

Reward is only affected by the agents' action choice in the current interaction. However, learning takes place via social pressure from repeated interaction, thus the history of interaction indirectly influences agent's action choice.

Delgado et al. [3] investigate a similar norm emergence scenario with several key differences. The agents in their research are restricted in their interactions to their neighbors in a scale free graph. Furthermore, their agents are playing a coordination game in which payoff is high if both agents chose the same action and low if both agents chose different actions. The authors formulate their action choice in terms of history. Each agent keeps a history of interactions and the corresponding reward. The agents then utilize the history to select the best payoff action. However, the history does not determine the reward they receive.

Kittock's research [4] is very similar to the research done by Delgado et al. summarized above. Kittock also utilizes the same style of payoff metric used in Delgado's work as well as using a graph to restrict agent interactions. Agents utilize memory of interaction payoffs to select their actions. His work is different in that he investigates several graph topologies and payoff matrices.

III. Model

The social learning situation for norm emergence that we are interested in is that of learning to reach a social convention. We borrow the definition of a social convention from [5]: *A social law is a restriction on the set of actions available to agents*. A social law that restricts agents' behavior to one particular action is called a social convention.

We represent the interaction between two agents as an n-person m-action game. At each time step, each agent is paired with another agent and decides in which state it wants to be. In our case, as in the case in [3], a social convention will be reached if all the n agents are in the same state, i.e., the actual state chosen is immaterial. For our purpose, an agent choosing a particular action is equivalent to it being in a corresponding state. In this paper, we consider only binary interactions, i.e., n=2, and agents are choosing between one of two available actions, A and B, i.e., m=2.

We consider the following three different environment types: (i) a *one-dimensional lattice* with connections between all neighboring vertex pairs (examples can be seen in Figures 1(a) and 1(b)); (ii) a *scale-free network*, whose node degree distribution asymptotically follows a power law (an example can be seen in Figure 1(c)); (iii) to further our understanding of the norm emergence process, and to capture some typical real-world scenarios, e.g., a community of closely knit researchers and their students, we use a rather novel network topology, namely the *fully connected stars network*: such a network has a relatively small number of hubs or core nodes which are fully connected forming a clique, and each of these core nodes is also connected with a number of leaf nodes (an example can be seen in Figure 1(d)).

Each agent is represented by a node in the network and the links represent the possibility of interaction between nodes (or agents). The *one-dimensional lattice* provides a structure in which agents are connected with their *n* nearest neighbors. Different values of the neighborhood size (*n*) produces different network structures; for example, when $n = 2$ the network will have a ring structure (as in Figure 1(b)) and agents will only be connected with their direct neighbors (those at left and right if we imagine a ring topology). On the other hand, when $n = PopulationSize$, the network is a fully connected network (as in Figure 1(a)) where each agent is connected with all other agents. On the other hand, in the *scale-free network* there are many vertices with small degrees and only few of vertices with large degrees. This makes the network diameter [1] significantly small with respect to the *one-dimensional lattice*.

As in [4], we use agents with a memory M_k of size M (same size for all the agents). For agent k, the memory M_k will record some information on the history of its decisions: The value of the position *i* of the memory M_k will be a tuple $\langle a_k^i, t^i \rangle$ where t^i is the time the *i*-th memory event took place, and a_k^i is the decision taken by agent *k* at time t^i ($1 \leq i \leq M$). Thus, the memory of each agent will work as a record of the history for the last *memory size* actions taken by the agent.

Agents cannot observe the other agent's memory, current decision, or immediate reward, and hence cannot calculate the payoff for any action before actually interacting with the opponent. When two agents interact, the instantaneous reward that an agent receives is calculated based on the action it selected and the action history of both agents as shown in Algorithm 1, where A_x and B_x are the number of *A* and *B* actions in memory that agent x has taken, $Action_x$ is the last action taken by agent x, and for which it is rewarded, $MajorityAction$ is selected to be whichever action is played most by the two playes combined, $MajorityActions_x$ is the number of actions in x's memory equal to the majority action, and $TotalMajorityActions$ is the number of times the majority action was chosen by both players in their finite histories.

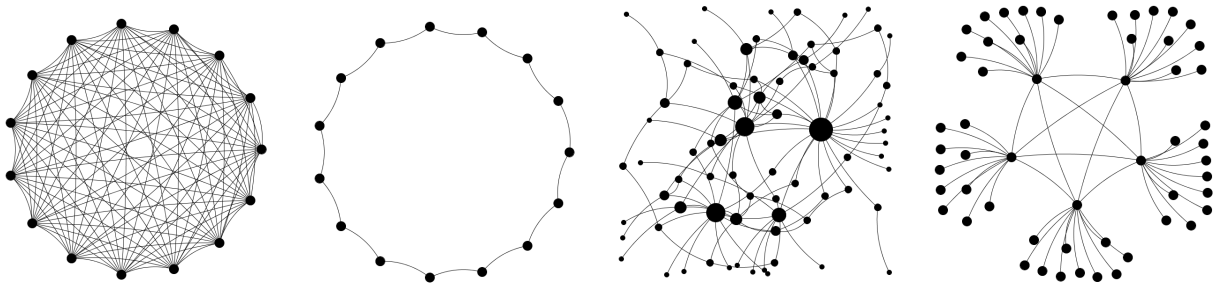
Agents use a learning algorithm to estimate the worth of each action. Agents will choose their action in each interaction in a semi-deterministic fashion. A certain percentage of the decisions will be chosen randomly, representing the exploration of the agent, and for the rest of the decisions, the agents deterministically choose the action estimated to be of higher utility. In all the experiments presented in this article, the exploration rate has been fixed at 25%, i.e., one-fourth of the actions are chosen randomly.

The learning algorithm used here is a simplified version of the Q-Learning algorithm [6]. The Q-Update function for estimating the utility on an action is:

$$Q^t(a) \leftarrow (1 - \alpha) \times Q^{t-1}(a) + \alpha \times reward \qquad (1)$$

where $reward$ is the payoff received from the current interaction and $Q^t(a)$ is the utility estimate of action a after selecting it t times. When agents decide not to explore, they will choose

[1]The diameter of a graph is the largest number of vertices which must be traversed in order to travel from one vertex to another



(a) Fully Connected Network (b) Ring Network or One-dimensional Lattice with Neighborhood Size 2 (c) Scale-Free Network (d) Fully Connected Stars Network

Fig. 1. Underlying Topologies

Algorithm 1: Memory Based Reward Function.

```
// First, we select the majority action
TotalAActions = A_1 + A_2;
TotalBActions = B_1 + B_2;
if TotalAActions < TotalBActions then
  | MajorityAction = B;
if TotalBActions < TotalAActions then
  | MajorityAction = A;
if TotalAActions == TotalBActions then
  | MajorityAction =
  | RandomlyselectedbetweenAorB;
// Then, we calculate the reward
   depending on the agents action
   selection and on the majority action
if Action_1 == MajorityAction then
  | reward_1 = MajorityActions_1 / TotalMajorityActions
else
  | reward_1 = 0
end
```

the action with higher Q value. The reward used in the learning process is a proportional reward of that calculated previously with Equation 1.

The simulation process for repeated interactions in the agent society is presented in Algorithm 2.

Algorithm 2: Simulation Process.

```
for timesteps do
  | forall agents do
  |   | Select another partner agent from population;
  |   | Each selected agent chooses an action;
  |   | The joint action from the selected agents and
  |   | their history determines rewards;
  |   | Selected agent(s) use received reward to update
  |   | action estimates;
  | end
end
```

We have used two different learning modalities: (a) In the *Multi learning* approach both interacting agents use the payoff to update their memory and policy, (b) In the *Mono learning* approach, however, only the first agent selected, and not the second one, updates its memory and action estimate after an interaction. Each agent interacts exactly once per time step in mono-learning, whereas in multi-learning different agents interact different times in the same time step because of randomness of partner selection.

IV. Experiments

To evaluate the rate and success of norm emergence we ran experiments with different societal configurations by varying the following system and agent properties:

- **Memory Size:** We vary the number of past interactions stored by an agent, so we can analyze the effects of memory sizes.
- **Population Size:** We study both the effects of population size and results that were not affected by scale of the population.
- **Neighborhood Size:** We study how different neighborhood sizes in a one dimensional lattice affect the process of emergence of conventions.
- **Underlying Topology:** We observe the dynamics of the process of emergence of conventions depending on the underlying topology.
- **Learning Modalities:** We compare how conventions are reached with different learning modalities, namely, one or both agents learning from an interaction.

Results reported here have been averaged over 25 runs. Agents are initialized with uniformly random memories, and initially are unbiased in their action choice. We conclude that a social convention has been reached when 100% of the population choose the same action. Other authors in the literature such [4] or [3] fixed the convergence rate at 90%. However we have observed that with certain reward functions on certain topologies, even after 90% of the society has converged to a convention, it can still switch back to the other convention. Though some aspects of results from our simulated agent society can be transferred to human situations (with additional mechanisms), our results are targeted towards

a better understanding of how to develop self-adaptive agent societies.

A. Effect of Neighborhood Size

To observe the effect of neighborhood size, we use a one-dimensional lattice (as scale-free networks and fully connected stars predetermine the neighbors for each node) and use a memory size of 5. Figure 2 shows a comparison of convergence times for different neighborhood sizes, measured as percentages of the population size, in a multi learning approach.

We can see that when increasing the neighborhood size, the convergence time is steadily reduced until it stabilizes after a certain neighborhood size. This effect is due to the topology of the network. When the one dimensional lattice has a small neighborhood size, on average, the diameter of the graph is high and therefore agents located in different parts of the network need a higher number of interactions to communicate their decisions or arrive at a consensus. It is also interesting to note that for smaller neighborhoods, larger populations converge much faster.

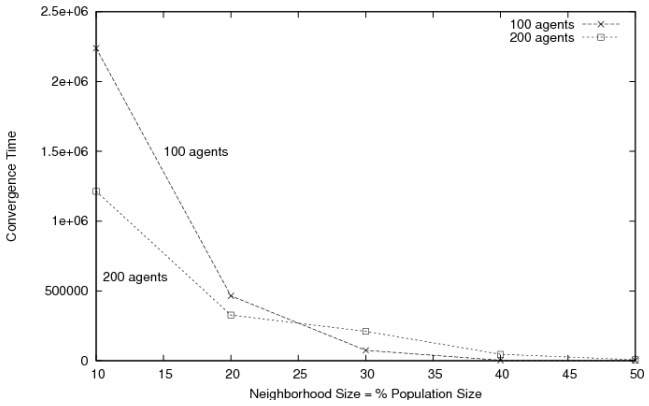


Fig. 2. Convergence rates with different Neighborhood Sizes in a One Dimensional Lattice (Multi Learning)

Similar convergence results are also obtained with the mono learning approach and we do not include them here due to space constraints. Once the neighborhood size crosses about 30% of the population size, the convergence time does not significantly decrease anymore. The relation of the neighborhood size and the diameter follows a geometric distribution and is shown in Figure 3. We note that when the neighborhood size crosses 30% of the population size, the diameter of the network is no longer significantly reduced, and hence the convergence times are also not significantly reduced any further.

B. Effect of memory size

In this experiment, we want to observe the effect of different memory sizes on convention emergence for different network topologies. We fix the population size at 100 agents. For the one-dimensional lattice, we use a fully connected network. We present the convergence times for different memory sizes in

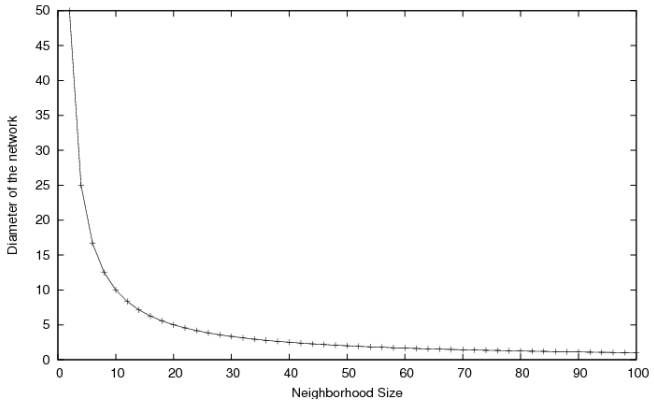


Fig. 3. Diameter relation with Neighborhood size in a One Dimensional Lattice with population = 100

Figure 4. The results show that larger memory sizes gradually increase time for convergence. This phenomenon is due to the configuration of the reward function and the learning algorithm. Each action in memory gets a relatively high reward for smaller compared to larger memory sizes (refer to the reward function defined in Algorithm 1). The learning algorithm, therefore, receives larger reinforcements for the actions performed for smaller memory sizes, resulting in faster convergence. Convergence is accelerated in this situation because higher rewards have a larger impact on the Q value updated by the learning algorithm 1. On the other hand, when dealing with higher memory windows, the proportional reward is much smaller, and therefore, the reinforcement will be smaller. Due to this smaller reinforcement, a higher number of interactions, and hence higher number of timesteps, will be needed to reinforce that action to same degree, thereby increasing convergence time.

We also note from Figure 4, that the mono-learning approach takes longer to converge than the multi-learning approach. A part of this difference is explained by the fact that the average number of learning interactions in a multi-learning approach is twice that of the mono-learning approach for the same number of time steps. There is, however, an additional clear trend of accelerated learning when both agents are learning from the same interaction.

In Figure 5 (note that the y-axis is in a logarithmic scale), we can observe the relative performance of different topologies for different memory sizes with the mono-learning approach. During this experiment, we limited the execution of the simulations to one million timesteps. We observe that the Fully Connected Stars network takes the most time to converge, followed by the Scale-Free network. For both the Scale Free and the Fully Connected Networks we can observe that the convergence time increases with increasing memory size. These inefficiencies are largely due to more time taken to break or resolve conflicting subconventions that form with scale-free and fully connected stars networks but not for fully connected networks (see following section for an explanation).

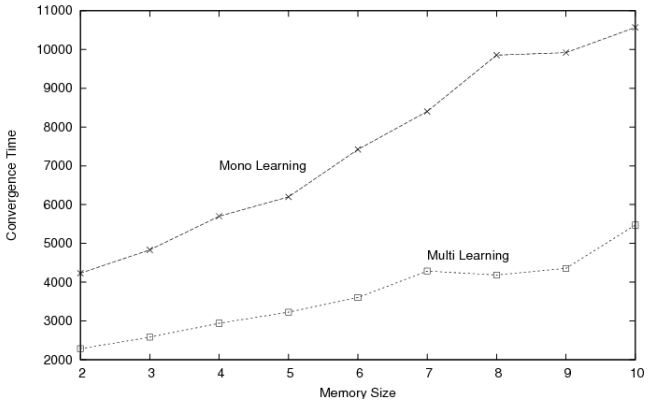


Fig. 4. Effect of Memory Size in Convergence Time on a Fully Connected network. (100 Agents).

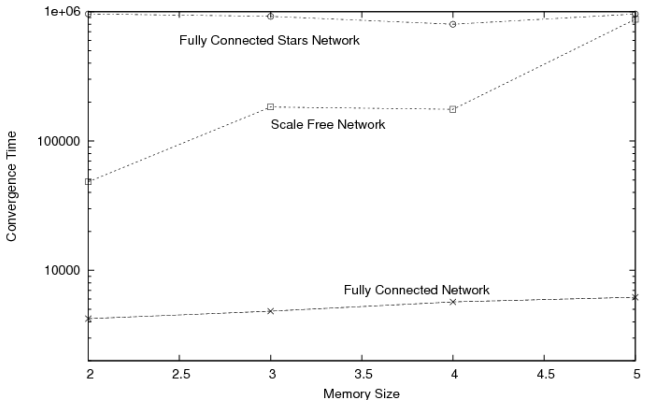


Fig. 5. Topologies Comparison with different Memory Sizes with Mono Learning Approach. (100 Agents).

As a result, the fully connected network, scales up much better with increasing memory size.

C. Effect of Learning Approach

In this set of experiments, we observe the difference in convergence times with the two learning approaches for different topologies. We first compare results of the two learning approaches in a one-dimensional lattice with 100 agents (see Figure 6, where the y-axis is drawn on a logarithmic scale). For smaller neighborhood sizes, i.e., when the network diameter is high, multi-learning takes longer to converge than mono-learning. After reaching the point where the diameter is no longer affected by the neighborhood size (as discussed before, this happens when the neighborhood size is about 30% of the population size) , the multi learning performs better. The reason for this interesting phenomenon is the creation of local subconventions with multi-learning when the diameter is large.

When agents have a small neighborhood size, they will interact often with their neighbors, resulting in diverse subconventions forming at different regions of the network. With the multi-learning approach, agents reinforce each other in each interaction. Such divergent subconventions conflict in overlapping regions. To resolve these conflicts, more interactions are needed between the agents in the overlap area between regions adopting conflicting subconventions. Unfortunately, the agents in the overlapping regions may have more connections in their own subconvention region and hence will be reinforced more often by their subconventions, which makes it harder to break subconventions to arrive at a consistent, uniform convention over the entire society. In the case of the mono-learning approach, the agents in the overlapping region will not be disproportionately reinforced by the other agents sharing its subconvention, making it easier to break those subconventions.

On the other hand, when neighborhood sizes are large, and hence network diameters are small, agents interact with a large portion of the population. This makes it more difficult to create or sustain subconventions. In addition, this large neighborhood size is more effectively utilized by the multi-learning as agents will be learning from all the interactions they are involved in, and not only from the interactions initiated by them.

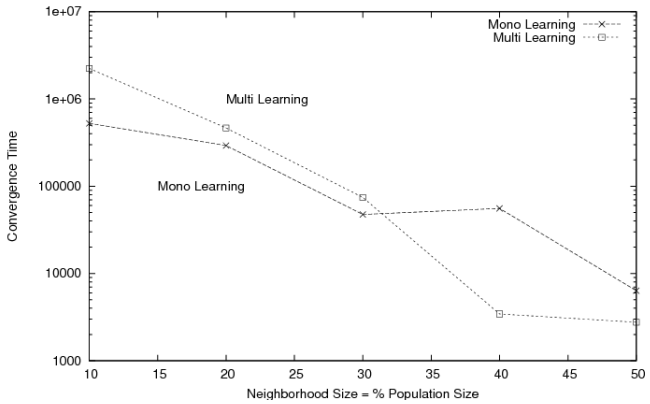


Fig. 6. Different Learning Approaches in One Dimensional Lattices with different Neighborhood Sizes

For the scale-free and fully-connected stars networks, systematic variation of neighborhood size is not possible in general. We do observe an interesting phenomenon for these kind of networks. When the multi-learning approach is used in Scale Free Networks and Fully Connected Stars, subconventions are persistent and the entire population does not converge to a single convention. This is the first time in all of our research on norm emergence that we observed the coexistence of stable subconventions.

The explanation of this rather interesting phenomena can be found in the combination of the memory-based reward function and the inherent topologies of such networks. We present, in Figure 7, a portion of a representative Scale Free or Fully Connected Stars network where subconventions have formed. We see that agent 1 (hub node 1) and its connected leave nodes (nodes 10, 11, and 12) have converged to one subconvention (represented by the color of the nodes) that is different from the subconvention reached by agent 2 (hub node 2) and its connected leave nodes (nodes 21, 22, and 23). As an agent has equal probability of interaction with any of its

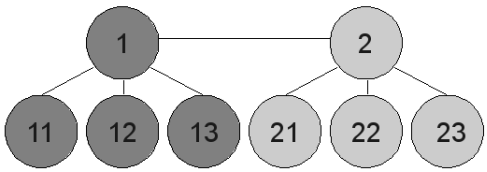


Fig. 7. Subnetwork topology resistent to subconventions in a multi learning approach

neighbors, both agents 1 and 2 interact more frequently with their associated leave nodes that share their subconvention.

Also note that when two agents interact and both actions have been taken equally often in their combined memories, as will be the case when agents 1 and 2 interact, the majority action will be selected randomly, giving advantage to one of the agents. Now, in this scenario, for the subconventions to be broken it is needed that for one of the hub nodes the following holds true: (1) the agent's q-value for its preferred action decreases, and (2) the q-value for the action preferred by the other agent increases. In order for *the agent's q-value for its preferred action to decrease*, a number of repeated interactions (proportional to the memory size) between the hub nodes (in our example 1 and 2) have to occur, and as there will be no clear majority action, the preference has to be given to the same action, e.g., that preferred by agent 2, in all those interactions. As the reward for the agent 1's action will then be 0, its q-value will start decreasing. In order for the agent 1's q-value for its non-preferred action to increase, a number of interactions (also proportional to the memory size) between it and agent 2 has to occur and agent 1 has to explore in that interaction and try agent 2's preferred action. This will result in agent 1's estimate of agent 2's preferred action to increase, albeit slowly. Only when both these fortuitous events follow each other, and without the intervention of another interaction with the leaves associated with agent 1 (which would reinforce the subconvention), can the subconvention be ultimately broken. The likelihood of these sequence of events happening is exceedingly small and hence subconventions routinely arise with the multi-learning approach. Viewed another way, the leaf nodes can only interact with their hubs and each of them will reinforce the subconvention action for their associated hub node in every time step, making it extremely difficult to resolve conflicting subconventions as in the situation in Figure 7.

On the other hand, as an agent is reinforced only once each time-step in the mono-learning approach, the processes required to break the subconventions are more likely, even though the corresponding probability is still relatively small. This probability decreases with larger memory sizes and hence subconventions are more likely to emerge with larger memory sizes when using mono-learning. There is a correlation with the memory size. Therefore, subconventions persist longer with larger memory sizes, and this phenomenon caused the significant convergence time for scale-free networks and full-connected star networks (we discussed this in the previous section with reference to results displayed in Figure 5).

D. Weighted Reward

To facilitate the reconciliation of subconventions, we decide to investigate a reasonable modification of the reward function. The current reward function only takes into account the previous actions chosen by both agents. In particular, the identify, or more specifically, the social position of the interacting agents did not influence the rewards calculated. We can, however, easily imagine scenarios where the position or social status of an agent can influence the payoff calculation. A straightforward way to incorporate social status in reward calculation would be to use a multiplicative weight, depending on the degree of the interacting node[2], in Algorithm 1 presented in section III. As a result, interactions with central, better connected nodes will produce higher rewards than those with relatively isolated nodes on the fringe of the network. By using this weighted reward we are allowing the hub agents to have a greater influence on other agents. The new reward function calculation is described in Algorithm 3.

Algorithm 3: Memory and Social Position Based Reward Function.

```
// First, we select the majority action
TotalAActions = A_1 + A_2;
TotalBActions = B_1 + B_2;
Weight = Degree_2;
if TotalAActions < TotalBActions then
  | MajorityAction = B;
if TotalBActions < TotalAActions then
  | MajorityAction = A;
if TotalAActions == TotalBActions then
  | MajorityAction =
  | RandomlyselectedbetweenAorB;
// Then, we calculate the reward
   depending on the agents action
   selection and on the majority action
if Action_1 == MajorityAction then
  | reward_1 = Weight × MajorityActions_1 / TotalMajorityActions;
else
  | reward_1 = 0;
end
```

In this algorithm, A_x and B_x are the number of *A* and *B* actions in memory that agent x has taken, $Action_x$ is the last action taken by agent x, $MajorityActions_x$ is the number of actions equal to the majority action that agent x has previously taken, $TotalMajorityActions$ is the number of actions of the majority action, and $Degree_x$ is the degree of agent x in the network.

Note that this modified reward function will not produce different results for the one-dimensional lattice networks, as all nodes in such networks have the same degree and hence will have the same multiplicative factor in the reward function.

[2]The degree of a vertex in a graph is number of edges connected to that vertex.

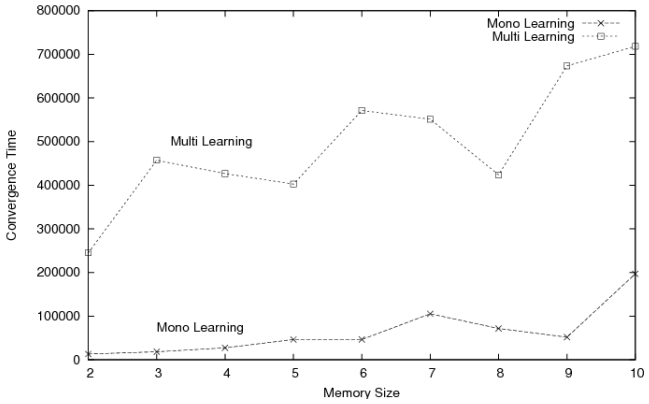


Fig. 8. Learning Approaches Comparison in Scale Free network with Weighted Reward Function.

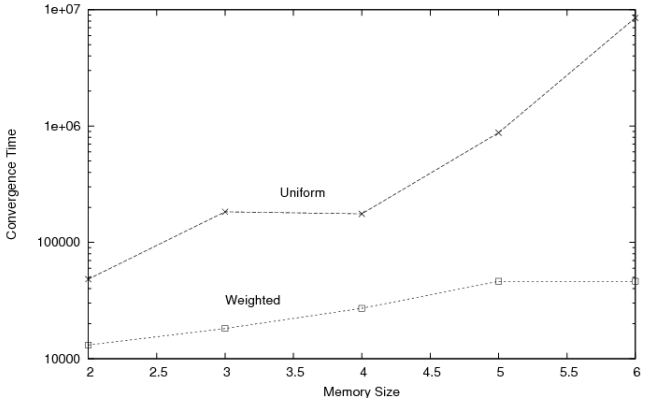


Fig. 9. Reward Functions Comparison in Scale Free network with multi-learning.

The results for the Scale Free network using the new Weighted Reward function and for the two learning approaches are shown in Figure 8. When we compare the results with this new Weighted reward function with those with the unweighted (uniform) reward function for Scale-Free networks with multi-learning (see Figure 9), we observe that the weighted reward function results in faster convergence. The main reason for this is that the weighted reward function allow the hub nodes to influence more weights and allows them to resolve subconventions (as leaf nodes have less influence on the hub nodes), and thereby producing faster convergence.

For the Fully Connected Stars networks, we observe that the Weighted Reward function produces faster convergence when using the mono-learning approach (see Figure 10). However, subconventions continue to persist with multi-learning approaches. The reason for this effect is due to the uniform degree distribution of the hub nodes in the network and the design of the reward function. The Fully Connected Stars networks engender a three phase convention emergence process: (1) first the leaf nodes drive the hubs, then (2) the hubs have to coordinate, and (3) finally the leaf nodes will have to coordinate with their hub. The second of these phases

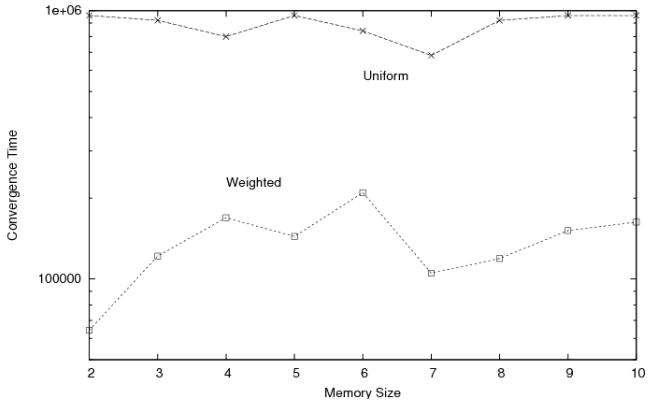


Fig. 10. Reward Functions Comparison in Fully Connected Stars network with mono-learning.

still takes significant time and is fragile, as explained in Section IV-C.

V. Conclusions

We have presented a set of experiments to study the emergence of social conventions based not only on direct interactions but also on the memory of each of the agents under different interconnection topologies between agents. This social learning framework requires each agent to learn from repeated interaction with anonymous members of the society. Norm emergence in real environments are likely to be influenced by both physical neighborhood effects imposed by mobility restrictions and biases as well as diverse learning, memory and reasoning capabilities of members of the society. Our main goal in this paper was to study the effects of these features on the rate of norm emergence.

Our initial hypotheses were that different characteristics of the topology in which agents are located would produce different convergence times for reaching a social convention. Experimental results confirm this hypotheses. We have shown that conventions emerge in less time when agents are allowed to interact with other agents located farther away from them in one-dimensional lattice topologies. The reason for this acceleration is that agents interact with a larger percentage of the population, which prevents formation of local conventions. We observe that memory size have a pronounced affetc on the emergence of conventions in all topologies studied with agents having larger memory sizes taking longer to reach conventions. This is due to the fact that the reward amount for a given action is inversely proportional to the memory size. As a result, reward sizes are smaller for larger memory sizes, requiring a higher number of interactions for a convention to be reached.

Finally, we have observed how the learning modality does directly affect the convention emergence process. We observe that subconventions are more likely to appear and are more resistant when using the multi-learning approach, and might not be resolved for scale-free and fully-connected star networks. To aid in breaking such stalemates, we introduced a new, plausible reward function which allows socially important

nodes (those with more connections to other agents) to have more influence in the reward function. This new reward function accelerates the emergence of conventions in scale-free networks but subconventions persist in fully-connected star networks.

VI. Future Work

One question that we plan to answer in future work is under what circumstances and configuration of parameters the one-dimensional lattice behaves similarly to the scale-free network for large population sizes. We have observed that when the population size increases, the convergence times in the one-dimensional lattice increases at a much faster rate compared to scale-free networks. We believe that a dynamic adjustment of the neighborhood size on a one-dimensional lattice will produce similar dynamics to those obtained with scale-free networks.

We also want to experiment with heterogeneous populations, as done by Mukherjee *et al.* [2]. In the current paper, all the agents are initialized with the same parameters and with the same distribution of initial memory. We want to observe the resulting dynamics of different types of populations. For example, in a scale-free network, we can initialize the hubs with a specific bias towards a certain action, and observe the speed of convergence of the rest of the population. Another interesting experiment to be carried out is when agents in the same population are initialized with different memory sizes.

Finally, to make the model more general, we want to increase the number of actions available to agents to $m > 2$. This extension will give us a more generalized game, and allow us to represent and study more diverse real-life situations.

Acknowledgment

This work was supported by the Spanish Education and Science Ministry [AEI project TIN2006-15662-C02-01, AT project CONSOLIDER CSD2007-0022, INGENIO 2010]; Proyecto Intramural de Frontera MacNorms [PIFCOO-08-00017] and the Generalitat de Catalunya [2005-SGR-00093]. Daniel Villatoro is supported by a CSIC predoctoral fellowship under JAE program. Sandip Sen is partially supported in part by a DOD-ARO Grant #W911NF-05-1-0285.

References

[1] S. Sen and S. Airiau, "Emergence of norms through social learning," *Proceedings of IJCAI-07*, pp. 1507–1512, 2007.

[2] P. Mukherjee, S. Sen, and S. Airiau, "Norm emergence with biased agents," *International Journal of Agent Technologies and Systems (IJATS)*, vol. 1, no. 2, pp. 71–84, January 2009.

[3] J. Delgado, J. M. Pujol, and R. Sangüesa, "Emergence of coordination in scale-free networks," *Web Intelli. and Agent Sys.*, vol. 1, no. 2, pp. 131–138, 2003.

[4] J. E. Kittock, "Emergent conventions and the structure of multi-agent systems," in *Lectures in Complex systems: the proceedings of the 1993 Complex systems summer school, Santa Fe Institute Studies in the Sciences of Complexity Lecture Volume VI, Santa Fe Institute*. Addison-Wesley, 1993, pp. 507–521.

[5] Y. Shoham and M. Tennenholtz, "On the emergence of social conventions: modeling, analysis, and simulations," *Artificial Intelligence*, vol. 94, pp. 139–166, 1997.

[6] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992. [Online]. Available: http://jmvidal.cse.sc.edu/library/watkins92a.pdf