Using Tags to Evolve Trust and Cooperation Between Groups

Aviv Zohar Jeffrey S. Rosenschein
School of Engineering and Computer Science
The Hebrew University of Jerusalem
91904 Jerusalem, Israel
{avivz, jeff}@cs.huji.ac.il

Abstract

Previous models of tags for the evolution of cooperation have demonstrated their effectiveness both in promoting cooperation and in allowing new strategies to take hold in otherwise incompatible populations. While previous work concentrated on interactions within a group of agents, this paper expands the use of tags to interactions between groups. In settings where an agent's tag is correlated with its behavior, it seems a reasonable heuristic to use stereotypes based on tags when interacting with "strangers".

We study a simple model of interactions between boundedly rational players playing the prisoner's dilemma, who are only allowed to see one another's tags prior to choosing an action. We consider the effects of various parameters on the level of cooperation achieved and on overall social utility: the number of interactions between groups, the number of game rounds, the number of memory states, the mutation rate of agents, and the size of the tag space. Computer simulations of our model show that with the tag mechanism in place, cooperation between different groups of players can become common. We also discuss some of the strategies that evolved in our simulations, and analyze interactions between quickly forgiving players.

1 Introduction

The emergence of cooperation in societies of selfish individuals has been the subject of much research in biology, sociology, mathematics, psychology, political science, and computer science. In real-world open systems, agents are free to act autonomously, and it is to be expected that each agent will act to maximize its gain, sometimes at the expense of others. The notion of social welfare in such systems may contradict individual welfare, and interactions among agents might be hindered by the greedy choices individuals make.

1.1 The Prisoner's Dilemma

One of the most common problems for exploring the tension between social and individual welfare is the prisoner's dilemma [1]. The prisoner's dilemma is a simple game between two players, each having two possible actions — **C**ooperate and **D**efect. The game's payoffs are usually named: **R**eward, **T**emptation, **S**ucker's payment, and **P**enalty. The following preference is defined over payments:
$T > R > P > S$ and $2R > T + S$.
These payoffs make defection more appealing than cooperation for each of the players, regardless of the strategy selected by the other. However, if both defect, a low payment results for both. For the purposes of this paper, the payoffs were set to the following values:
$T = 4 \quad ; \quad R = 2 \quad ; \quad P = -1 \quad ; \quad S = -2$.

The prisoner's dilemma has also been widely explored in its iterated version (IPD). Axelrod [1, 2] demonstrated through competitive simulations that, unlike the single game version, cooperation in the repeated game may be successful because of the looming "shadow of the future". The strategy that prevailed in Axelrod's tournament and evolutionary simulations was a surprisingly simple one: Tit-For-Tat. This strategy begins with cooperation, then simply echoes the opponent's previous move. Thus when the opponent cooperates he is quickly rewarded with cooperation, and when he defects he is quickly punished with defection.

Besides the IPD tournament and the evolutionary setting, other models of learning effective strategies have been explored. For example, the repeated prisoner's dilemma has also been extensively researched as a reinforcement-learning problem [13].

1.2 Evolutionarily Stable Strategies

Tit-For-Tat did not always emerge as the dominant strategy in Axelrod's evolutionary simulations. Other strategies, such as the strategy of always defecting (ALL-D), are stable in the population and resist invasion.

An Evolutionarily Stable Strategy (ESS) [10, 9] is a mixture of strategies for the members of a population such that if most of the population is following the ESS, then any player that deviates from it would do worse than the others. In an evolutionary setting, this means that the deviating player will reproduce less than the others and will eventually become extinct. The population thus maintains this mixture of strategies over time. However, there may be several stable mixtures of strategies (or none at all) and the equilibrium that the population converges to is influenced by chance and the initial set of strategies. A mixture of strategies will be called a weak ESS if deviating from it is not strictly beneficial to the deviating player. Tit-For-Tat is a weakly stable strategy, and so is ALL-D.

One way a group of strategies can invade an ESS is by remaining isolated from the rest of the population. Niches are isolated environments where the population is allowed to evolve independently. The existence of niches maintains a large diversity in an evolving population since the evolutionary paths in separated niches may develop in entirely different ways. In a different niche, the population might converge to a different ESS.

1.3 Tags

Tags (also called "labels" by some) are genetically determined markers, which are visible to all. When two agents interact, they can both see each other's tags and use that information when deciding how to act. The idea behind tags is that players might choose to play only with players that have a tag sufficiently similar to their own. Since tags are also genetically encoded, having the same tag as some other player implies that other genes may also be present in both players. Tags also allow for an artificially imposed separation of players into nearly isolated groups — very much like niches.

The tag mechanism is also very closely related to kin selection. Dawkins [3] gives the example of the "green beard gene" that allows its carriers to recognize one another while at the same time inducing them to help one another. On the other hand, Hales has demonstrated in [5] that tags are also useful in situations when the tag-group is composed of non-kin.

Riolo [11] uses tags with the iterated prisoner's dilemma and shows that they help reach cooperation faster. In [8, 12] tags are shown to offer an alternative to repeated play for establishing cooperation. A tag system with very simple strategies that is used in one-shot interactions is shown to achieve cooperation. An important characteristic of these systems is that they display fluctuations in tolerance due to the invasion of exploiters into cooperative groups. See, for example, [14] for a mathematical model of this effect.

The tag mechanism described above is very appealing due to its simplicity and low cost. All that is needed is that each player hold a constant identifying label and show it to potential partners. No memory of interactions or complex reasoning is needed.

Tag models that only allow for interaction between similar individuals, however, may be problematic in some settings. Agents might not be able to afford to be too choosy as to whom they interact with — they might be missing out on a lot of potentially beneficial interactions. On the other hand, the power of the tag mechanism lies in its simplicity. Players are able to make choices quickly and without spending too much effort on finding the best strategy.

1.4 Extending the Model of Tags to Include Interactions Between Groups

In this paper we explore an expanded model of tags that involves interactions between groups, and explore through simulations and analysis its properties. In Section 2 we describe the model that was explored and its relation to the repeated play and tag models. We then consider the effects of various parameters on the level of cooperation achieved and on overall social utility: the number of interactions between groups (Section 3.1), the number of game rounds (Section 3.2), the number of memory states (Section 3.3), the mutation rate of agents (Section 3.4), and the size of the tag space (Section 3.5). Computer simulations of our model show that with the tag mechanism in place, cooperation between different groups of players can become common. We also discuss some of the strategies that evolved in our simulations in Section 3.6, and analyze interactions between quickly forgiving players in Section 4.

2 The Model

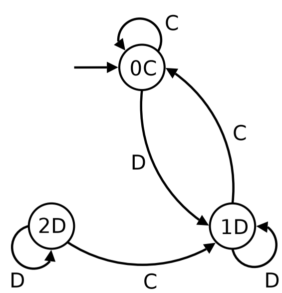
In our model, a population of players is allowed to interact by playing one-shot prisoner's dilemma games. At every round, players are randomly paired for a game. Every player's genes encode a tag from a discrete tag space T that is revealed to every partner it plays against and remains constant throughout its lifetime. We shall denote the set of actions available to each player at every round by $A = \{C, D\}$. The action choice for each game may be influenced by the tag of the opponent, and different opponents with the same tag cannot be distinguished from one another.

Each player is endowed with memory cells — one cell for each tag in the population (although we will see below that a much smaller number of memory cells is actually needed). We shall denote the memory cell of player i with regard to tag t as mem_i^t, and the set of states that each cell can be in as S. The state of the individual's memory determines which action it will choose when encountering another player, via the action selection function: $Action : S \rightarrow A$

After every game, the memory of both players is modified according to the action of their opponent. The change in memory of a player at any state is encoded in its genes and is represented by an update function: $MemUpdate : S \times A \rightarrow S$. An agent uses the same function for updating each memory cell — thus, the update function and action selection functions encode a meta-strategy that is consistent when playing different tags (but the actual game strategy of defection or cooperation can still vary between tags because of different experiences).

An easy way to present this meta-strategy is by drawing the automaton that matches it. As an example, consider Figure 1, which shows an automaton that encodes the Tit-For-Tat strategy. Each of the states is labeled with a number that is the representation of that state in memory, and with an action of C or D (for cooperation or defection) which is to be the next action played if the automaton is in that state. The two directed edges leading from every state indicate the change of state in response to experiencing cooperation or defection, according to the $MemUpdate$ function. The initial state (0 in this example) is marked by an arrow.

Figure 1: A 3-state automaton that encodes the Tit-For-Tat strategy



Note that the automaton in Figure 1 has some redundancy. State 2 can never be achieved since it is not the initial state, and there are no transitions leading into it. This way, an individual's genes may encode smaller automatons with fewer effective states than the maximum allowed number of states.

The gene representation of an automaton with N states contains:

1. N genes to encode the strategy played at each state (either Cooperate or Defect);

2. 2N genes to encode the reaction (memory update) at each state for each possible strategy played by the adversary (where each gene can have N different values);

3. 1 gene to encode the initial state of the automaton (also N possible values).

This gives a total of 3N+2 genes per individual in the population when including the gene that encodes the tag.

The formal description of a game encounter in which player i encounters player j that has tag $t \in T$ is given below (from the viewpoint of player i):

1. Observe j's tag t;
2. Play $Action(mem_i^t)$;
3. Observe j's action $a \in A$ and receive a reward;
4. Set $mem_i^t \leftarrow MemUpdate(mem_i^t, a)$.

At every generation, players get to play several such rounds of games during which their memory states are allowed to change. The fitness of players is then determined according to their average reward. Next, 10% of the population is randomly selected and paired up. The player with the higher fitness from each pair is then reproduced asexually and its offspring replaces the player with the lower fitness. During reproduction there is a small (around 1%) probability of mutation for each gene. When a mutation occurs, the gene is set to a new value which is chosen randomly from a uniform distribution over the allowed range.

2.1 A Comparison with Previous Models

The model above exists somewhere between the models of repeated play and the tag model that Hales and Edmonds explored in [8].

Similar to the repeated play model, our players also possess a memory of the actions of their opponents, and update this memory during play. The difference is that in our model, players cannot differentiate between members of the same group, and reciprocation therefore becomes harder — actions might not be directed at the right player. Just like in other tag-based models, we use a gene to encode each player's tag, which makes it somewhat coupled with the player's behavior. However, in our model players interact with others who are outside their tag group and are not likely to be kin. In fact, most of the interactions are between different tag groups, as is to be expected in a population composed of many groups.

3 Simulation Results

Throughout this paper we use the average reward per player per game as the measure of social utility. The highest attainable value is 2.0 which is achieved only if all players cooperate in all games. The lowest possible value is -1.0 which is achieved when all players defect. Figure 2 shows the social utility of a randomly initialized population of players.

After a transient phase at the beginning of the run, the system usually maintains a high degree of cooperation. The oscillations during the run emanate from the invasion of defectors into groups of cooperating agents. These defectors are initially doing better than other players, but as the number of cooperators in their group shrinks, these defectors, having killed off exploitable victims and having established their "reputation" as exploiters, earn less and are displaced by other tag groups. The resulting oscillations decrease in magnitude as the size of the population grows. This is to be expected in a randomly fluctuating system where fluctuations are "averaged out" as the system grows.

3.1 The Number of Interactions Between Groups

If the population only contains a few groups, and opponent-pairing is uniformly distributed, many of the games will take place between members of the same group. It is thus important to quantify how much of a player's fitness is determined by interactions with outsiders.

Figure 3 shows the portion of games that were played with members of the same tag group for a population of 1500 players with 5 memory states, a mutation rate of 0.01, and a tag space of 4000, playing 350 game rounds per generation.

At the beginning of the simulation, the population is initialized randomly and is spread out over a large portion of the tag space. This means that very few games are with members of the same tag group. Quickly however, one or two groups grow to hold most of the members of the population. Players in these large groups are very likely to meet other members of their group in random encounters, so many of the games are within the group. After a few generations, the population converges to a state where there are several medium sized groups. The population

Figure 2: The social utility in a typical simulation of the model

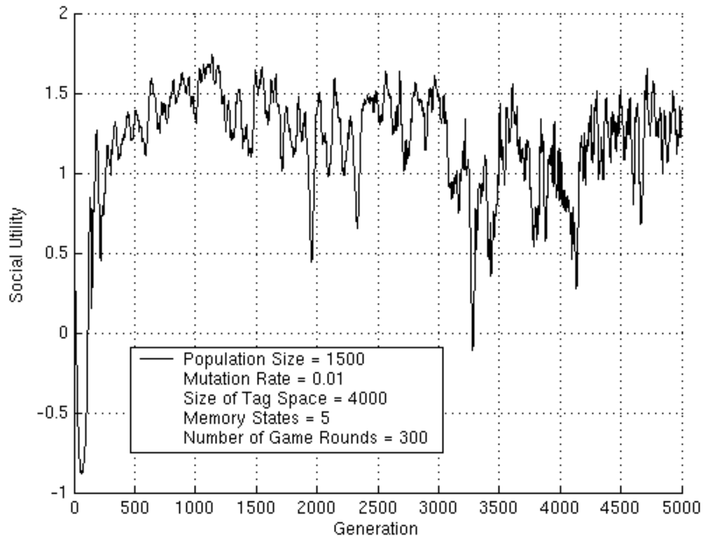
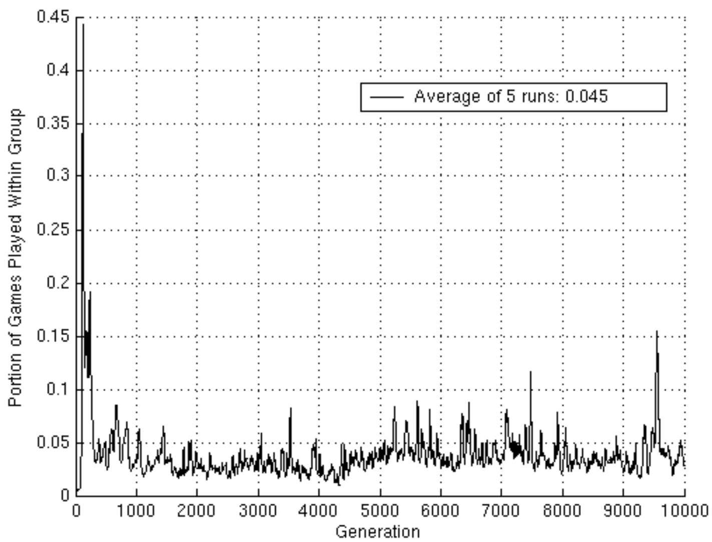


Figure 3: Portion of games played within the same tag group during a typical run

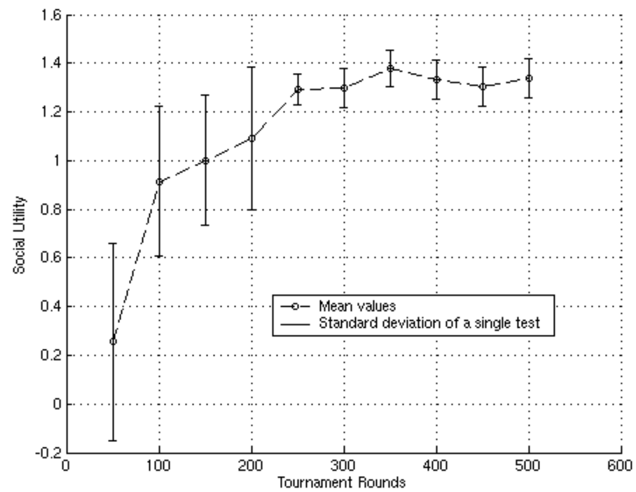


maintains that balance (although as the graph shows, there are oscillations in group size), and only around 4.5% of the games are played with members of the same group. This means that the most important factor in determining the fitness of players is how well they do when playing with dissimilar players. In view of this fact, it is encouraging to see the high levels of cooperation that were achieved.

3.2 The Effect of the Number of Game Rounds on Social Utility

Next, the effect of adding more game rounds at every generation was studied. It is to be expected that if more game rounds are played every generation, individuals that make use of their memory of past interactions will do better — this will support reciprocating players who may need several interactions with each group to determine with which groups they can cooperate. As seen in Figure 4, this has been found to be true. The social utility is usually higher if more game rounds are allowed in each generation. The tests were conducted on randomly initialized populations of 1000 players with 5 memory states and with a mutation rate of 1%. The simulations ran for 10000 generations.

Figure 4: Tournament rounds each generation vs. social utility



It is remarkable to observe that the system does indeed reach very high levels of cooperation. A social utility score of 1.4 is reached (from the possible range of [-1,2]). Larger systems with more games have been observed at levels of over 1.5 social utility, where in 75%–80% of all games both players chose to cooperate.

3.3 Increasing the Number of Memory States

Next, the effect of endowing the players with more memory states was explored. Giving players more memory would not necessarily benefit only the "nice" players. It may also benefit the exploiters, who will be able to better evaluate when to defect.

Tests were run for populations of players with different numbers of memory states. Each of the tests ran for 10000 generations on a population of 1000 individuals with a mutation rate of 0.01 and a tag space of 4000. The results appear in Figure 5.

It appears that adding more memory to players raises the social utility, but that more than 4 memory states are not very useful. Defectors were usually found to use very simple strategies that did not make use of all the available states while cooperators were usually more complex. Since players can evolve automatons with unreachable states, there does not appear to be any obvious disadvantage to having more memory states.

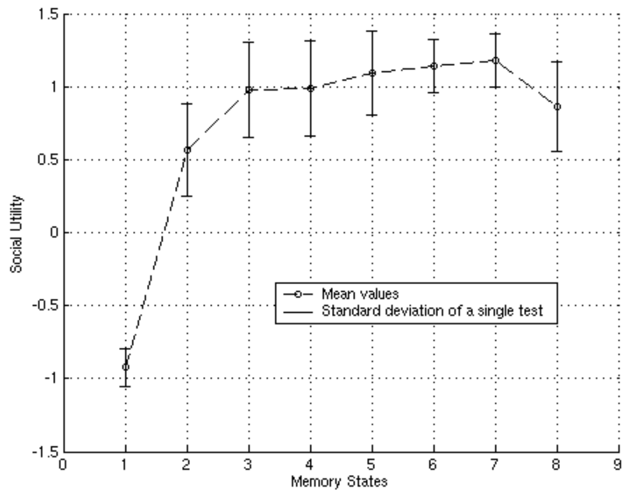
Note that if the players are only allowed 1 memory state, they are playing a fixed strategy. In this case, the population always converged to a population of defecting players. Two memory states are already enough to encode the Tit-For-Tat strategy, but here as well, the population did not always converge to high values (the large variance indicates that there were very large differences between various tests).

3.4 The Effects of Varying the Mutation Rate

In some of his recent research, Hales suggests that for a tag-based system to work well, tags must change quickly compared to the rate at which mutations occur [6]. That way, cooperative groups spawn more cooperative groups faster than exploiters destroy them from within. The mutation rate that we have applied is the same for both the tag and other genes.

It would appear at first glance that since behavior is governed by many genes, each of which is susceptible to mutation with the same probability as the tag, the condition Hales requires does not hold in this model — especially

Figure 5: Number of memory states vs. social utility



when the number of memory states is large (which means that there are more genes that might suffer mutation). In fact, for a player with 5 memory states, a mutation rate of 0.01 actually gives a very high chance of an error in reproduction. Only around 84% of the offspring of an individual will be copied without any error in one of the 17 genes that encode their behavior and tag. However, since automatons may contain many redundant states, the agents in this model may evolve to use a smaller portion of the states, thereby decreasing the chance of mutation to the automaton compared to that of the tag (mutations on genes pertaining to unreachable memory states do not matter since these mutations cannot make those states reachable). The proportions between the mutation rates can thus be modified by evolution itself.

Figure 6 shows the results of runs of the simulation with various mutation rates. All tests were run for 10,000 generations on a population of 1000 individuals with 5 memory states, playing 250 rounds of games per generation and a tag space of 4000.

At a mutation rate of zero, the population converges to complete cooperation, since there is a very good chance that there is a player within the randomly initialized population that plays a variation of Tit-For-Tat. Since there are no possible invasions by exploiting individuals, these players manage to outperform other strategies and reproduce. Adding a small mutation rate allows for some small probability of invasion into each tag group, and so the social utility declines. This tends to repeat itself consistently over the tests that were taken for each data point (there is a relatively low variance between the tests, as the graph indicates).

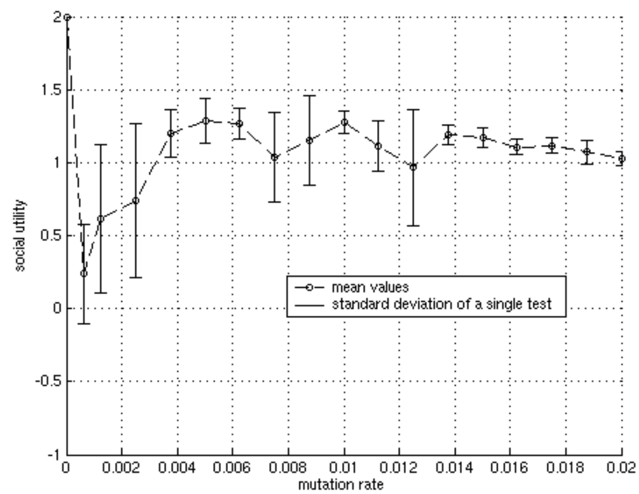
At 0.001 mutation rate, the average social utility over the tests is low. In some tests there was no emergence of cooperation at all (or rather, mutations were frequent enough to allow invasion to eliminate most emerging groups of cooperators); thus a high variance was found among the tests.

Increasing the mutation rate above 0.001 brings about a rise in the social utility of the system. This is probably where the mutation rate of tags becomes large enough to allow groups of cooperators to expand quickly to unoccupied regions of the tag space. As the mutation rate is increased further towards a much higher rate, the system degrades very slowly, possibly because of the loss of correlation between tag and behavior that higher mutation rates cause.

3.5 The Size of the Tag Space

A large tag space has been previously identified [4] as a factor contributing to cooperation. Tag-based systems contain a never-ending race between cooperative players and exploiters. When exploiters invade a group of cooperative players and reproduce within it, some of their offspring may end up with mutated tags and may invade other groups in turn. This is very much like the behavior of viruses and epidemics. If the tag space is sufficiently large, it would be very sparse, and the probability of a mutated tag "hitting" an occupied tag value becomes very small. The appearance

Figure 6: Effects of mutation rate on social utility

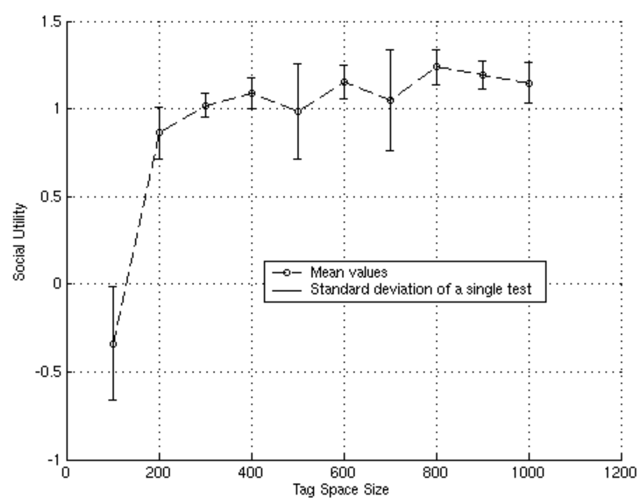


of exploiters is then governed mostly by mutations to the individuals already within the group.

Another benefit of a large tag space is that it allows “nice” groups a chance to form even in an ALL-D population. Mutations are more likely to fall on an empty tag and may start to evolve and interact with some separation from the rest of the population.

The results displayed in Figure 7 support these hypotheses and demonstrate that a larger tag space allowed for more cooperation on average in the system.

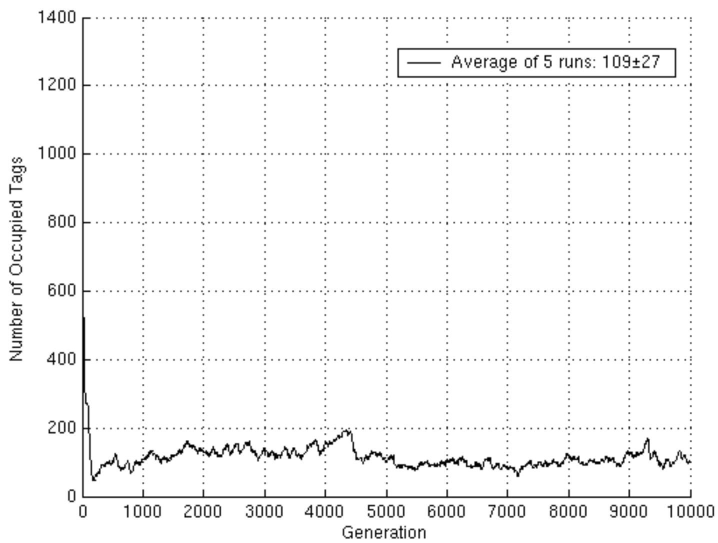
Figure 7: Size of tag space vs. social utility



The problem with large tag spaces is that we have assumed each player has a memory cell for each tag in the tag space. The cost of maintaining such a large memory for each player may be high. However, in reality, only a small portion of the tag space is occupied, and each player uses very few memory cells.

The results shown in Figure 8 demonstrate this point. The population contains 1500 players with 5 memory states, a mutation rate of 0.01 and a tag space of 4000 playing 350 game rounds per generation.

Figure 8: Usage of tag space during a typical run



When the system is initialized, there are extremely high numbers of occupied tags — almost 1 tag for every player in the population. This is due to the random initialization of the population. Very quickly, the number of occupied tags drops and is maintained around 109. This is less than 1/10 the size of the population (which was 1500 in this run). In fact, some of the tag groups are very small, and a smarter memory allocation scheme may avoid allocating them memory without too much loss of utility to the player.

3.6 Some of the Strategies that have Evolved

As seen above, players with 2 memory states were less successful in evolving cooperative groups than players with more memory states. Since players with two memory states are complex enough to encode the Tit-For-Tat strategy, it appears that other strategies, which are more complex, evolve in the population and achieve a better rate of success.

Figure 9 presents a small selection of the automatons that evolved during a run of the simulation with players that have 5 memory states. These are not necessarily a representative sample of the population. An exact classification of the various mixtures of evolved automatons in the population is beyond the scope of this paper.

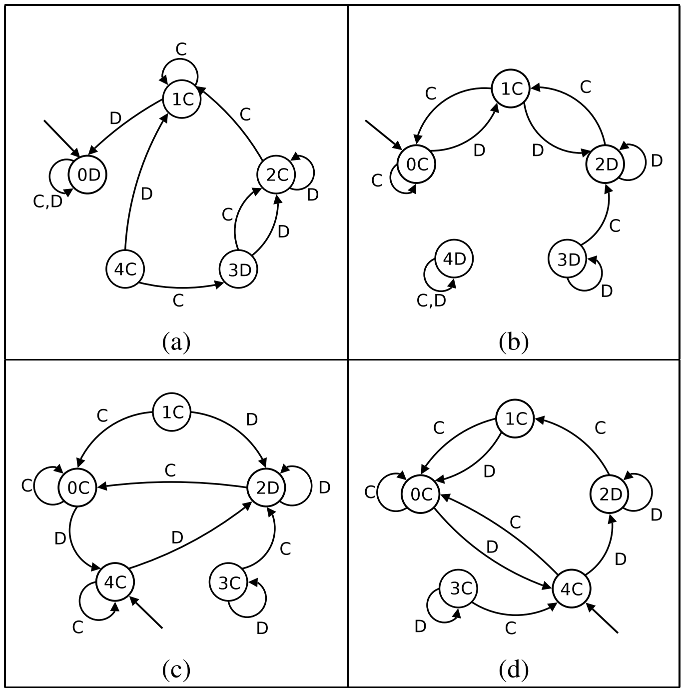
Automaton (a) is an All-D player that evolved after 650 generations and invaded the biggest tag group at the time. Note that the automaton for this player contains only a single relevant state. The memory is initialized to state 0 which is a defecting state and remains there no matter what strategies the opponents play. Automatons similar to this, as well as automatons that encoded strategies that defect most of the time, were frequently found during runs at times when social utility was declining.

Automatons (b), (c), and (d) belong to three players that evolved at different stages of the same simulation. All three play a variation of Tit-For-Tat and were taken from the largest tag group of their generation, where they were the most commonly found genome.

It is interesting to see that all three players play a more forgiving variation of Tit-For-Tat. They all initialize their memory to a cooperative state, and respond to many defections or many cooperations with the same move. However, they are usually forgiving towards single defections. It is quite surprising to see that these more forgiving versions of Tit-For-Tat are such good survivors since they are much easier to exploit. Their success might be explained by their increased resistance to noise. For a forgiving strategy, a single defection from a largely cooperating group will not trigger an escalation of retaliatory defections too easily (see [2] for more details).

In our case, noise is always a part of the system. It is caused by the mutations of single individuals in the group, which may cause them to deviate from the group's most common strategy. It is possible that 2 tag groups would be

Figure 9: Evolved Automatons

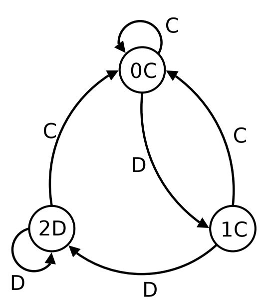


mutually defecting, while both are actually playing “nice” reciprocating strategies. Below, we show analytically how forgiving groups manage to overcome these situations.

4 Analyzing the Interaction Between Groups of Quickly Forgiving Players

A simple model can easily demonstrate the beneficial properties of a forgiving version of the Tit-For-Tat strategy. We show this for the automaton depicted in Figure 10.

Figure 10: A simple forgiving variant of Tit-For-Tat



If the player is at state 0, it takes 2 defections in a row to get it to retaliate. Forgiveness, however, is instantaneous and total. Experiencing cooperation from the opposite player returns the automaton to state 0 immediately. This is one of the versions of automatons that evolved during runs of the tag simulation.

We assume that there are two distinct groups of players, each with mixed proportions of individuals at the various memory states. For simplicity, both groups are assumed to be of the same size, and it is assumed that games are conducted synchronously between randomly selected pairs of players, one from each group.

Let us denote by p_s^t the proportions of players at memory state s at time t from one group of players, and by q_s^t the proportions of players from the other group.
$p_0^t + p_1^t + p_2^t = q_0^t + q_1^t + q_2^t = 1$

We can then derive equations to describe the dynamics of the system in the average case. For example, the expected proportion of players in state 0 in one group will be equal to the proportion of cooperating players in the other group in the previous time step — since every player who meets a cooperating player will move to state 0. We thus get:

$$\begin{array}{ll} p_0^{t+1} = q_0^t + q_1^t & q_0^{t+1} = p_0^t + p_1^t \\ p_1^{t+1} = p_0^t q_2^t & q_1^{t+1} = q_0^t p_2^t \\ p_2^{t+1} = (p_1^t + p_2^t) q_2^t & q_2^{t+1} = (q_1^t + q_2^t) p_2^t \end{array}$$

The change in the proportion of cooperative members of both groups is then:

$$cooperators^{t+1} - cooperators^t = \frac{p_0^{t+1} + p_1^{t+1} + q_0^{t+1} + q_1^{t+1}}{2} - \frac{p_0^t + p_1^t + q_0^t + q_1^t}{2}$$

Then using the state transition equations:

$$\begin{aligned} cooperators^{t+1} - cooperators^t &= \\ \frac{(q_0^t + q_1^t) + p_0^t q_2^t + (p_0^t + p_1^t) + q_0^t p_2^t}{2} &- \\ \frac{p_0^t + p_1^t + q_0^t + q_1^t}{2} &= \\ \frac{p_0^t q_2^t + q_0^t p_2^t}{2} &\geq 0 \end{aligned}$$

This means that the expected number of cooperating players is non-decreasing over time. Cooperation between the two groups will strictly increase within two rounds unless both groups are already fully cooperating, or both are composed entirely of players in the defecting state. Since the simulation we are modeling is noisy and stochastic, small perturbations will unbalance the equilibrium in the second case. The only stable state for the system is that of full cooperation. A similar analysis was conducted for several other forgiving variations of Tit-For-Tat with similar results. In fact, these results are applicable for more complex models, including asynchronous models.

5 Conclusions and Future Work

We have seen that the dynamics induced on a population by the tag mechanism allow for cooperation between groups through reciprocation. The cost of adding tags in the model presented is not high, and only a small amount of memory is required for each player. Players were modeled with simple decision-making mechanisms under bounded rationality assumptions, which can be easily implemented in a real-world setting.

As in previous work, the basis for cooperation in this model was reciprocation. Variations of Tit-For-Tat evolved in the system and were found to be frequent in the population. These individuals were usually found to be more forgiving than classic Tit-For-Tat, a property that we believe helps prevent an escalation in defections between two groups of players. As expected, we saw that adding to the computing power and the memory of players as well as allowing them more opportunities to learn about opponents (i.e., more game rounds per generation) supported more cooperation in the system.

Many directions for research still remain. The model we presented could be extended in the spirit of reputation systems and norm propagation to include information sharing between the agents, which may further increase cooperation. Another interesting direction for future work is to try and apply tags outside of an evolutionary setting. A crucial part of the tag mechanism is the high correlation between the behaviors of similar agents. Will tags still work if there is no genetic coupling between tag and behavior? Recent work by Hales [7] explores the use of tag-like mechanisms in peer-to-peer systems where the challenges of defining what a tag group is and how its members' behaviors are correlated seem to touch upon this very problem.

References

[1] R. Axelrod. *The Evolution of Cooperation*. Basic Books, New York, 1984.

[2] R. Axelrod. *The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration*. Princeton University Press, 1997.

[3] R. Dawkins. *The Selfish Gene*. Oxford University Press, New York, 1976.

[4] D. Hales. Cooperation without memory or space: tags, groups and the prisoner's dilemma. In S. Moss and P. Davidsson, editors, *Multi-Agent-Based Simulation*, number 1979 in Lecture Notes in Artificial Intelligence. Springer-Verlag, Berlin, 2000.

[5] D. Hales. Cooperation and specialization without kin selection using tags. CPM Working Paper 02-88, The Centre for Policy Modelling, Manchester Metropolitan University, 2002. http://cfpm.org/cpmrep88.html.

[6] D. Hales. Change your tags fast! — a necessary condition for cooperation? In *MABS'04 workshop at AAMAS'04*, New York, 2004. http://cfpm.org/ david/.

[7] D. Hales. From selfish nodes to cooperative networks — emergent link-based incentives in peer-to-peer networks. In *IEEE P2P2004*, pages 151–158, Zurich, Switzerland, August 2004.

[8] D. Hales and B. Edmonds. Evolving social rationality for mas using "tags". In *AAMAS'03*, pages 497–503, Melbourne, Australia, July 2003.

[9] J. Hofbauer and K. Sigmund. *Evolutionary Games and Population Dynamics*. Cambridge Univ. Press, Cambridge, 1998.

[10] J. Maynard-Smith. *Did Darwin Get it Right? Essays on Games Sex and Evolution*. Penguin Books, New York, 1993.

[11] R.L. Riolo. The effects and evolution of tag-mediated selection of partners in populations playing the iterated prisoner's dilemma. In *ICGA'97*, pages 378–385, 1997.

[12] R.L. Riolo, M.D. Cohen, and R. Axelrod. Evolution of cooperation without reciprocity. *Nature*, 414:441–443, 2001.

[13] T. Sandholm and R. Crites. Multiagent reinforcement learning in the iterated prisoner's dilemma. *Biosystems, Special Issue on the Prisoner's Dilemma*, January 1995.

[14] A. Traulsen and H.G. Schuster. A minimal model for tag-based cooperation. *Phys. Rev. E.*, 68, 2003. 046129.