Achieving Cooperation in a Minimally Constrained Environment

Steven Damer and **Maria Gini**

Department of Computer Science and Engineering
University of Minnesota
{damer,gini}@cs.umn.edu

Abstract

We describe a simple environment to study cooperation between two agents and a method of achieving cooperation in that environment. The environment consists of randomly generated normal form games with uniformly distributed payoffs. Agents play multiple games against each other, each game drawn independently from the random distribution. In this environment cooperation is difficult. Tit-for-Tat cannot be used because moves are not labeled as "cooperate" or "defect", fictitious play cannot be used because the agent never sees the same game twice, and approaches suitable for stochastic games cannot be used because the set of states is not finite. Our agent identifies cooperative moves by assigning an attitude to its opponent and to itself. The attitude determines how much a player values its opponents payoff, i.e how much the player is willing to deviate from strictly self-interested behavior. To cooperate, our agent estimates the attitude of its opponent by observing its moves and reciprocates by setting its own attitude accordingly. We show how the opponent's attitude can be estimated using a particle filter, even when the opponent is changing its attitude.

Introduction

Cooperation plays an important role in evolution, but it is difficult to explain how cooperation developed, since natural selection favors defectors who take advantage of cooperators without paying any cost. One of the mechanisms postulated for evolution of cooperation is direct reciprocity (Trivers 1971), where in repeated encounters two individuals can choose to cooperate or defect. This was formalized in the Iterated Prisoner Dilemma (Axelrod 1984) and in the Tit-for-Tat strategy, a strategy which starts with cooperation and then reciprocates whatever the other player has done in the previous round.

Cooperation can be valuable, but it can also be risky. When should an agent cooperate? How can an agent avoid exploitation? The risk of cooperation is that the opponent may not work to pursue a collective good, but instead take advantage of cooperative actions. This paper presents an environment where cooperation of agents is possible and beneficial. A unique feature of the environment is that it does not provide repeated exposure to a single game but only repeated interactions with the same player. The set of agents is

limited (we are currently using just two agents), so an agent has the opportunity to reciprocate the cooperative actions of the other agent. Each interaction is unique since the agents play a different game each time.

As mentioned earlier, it is possible to achieve cooperation in repeated games using Tit-for-Tat (Axelrod 1984) or variants such as win-stay lose-shift (Nowak & Sigmund 1993), but Tit-for-Tat requires that moves are labeled as cooperative or uncooperative. This is not suitable for agents that operate in environments that are too large and complex to be analyzed and labeled beforehand.

In order to create an environment suitable for cooperation and general enough to be adaptable to different situations we considered a number of criteria:

1. Cooperation must be possible. This excludes environments that consist of fixed-sum games where a gain for one agent is necessarily a loss for the other.
2. Exploitation must be possible as well - if there is no danger of exploitation then methods of cooperation might not guard against it, which would make them unsuitable for environments in which exploitation is possible. This excludes environments where agent's interests are completely aligned, such as if their payoffs are identical.
3. The opportunity for reciprocation is necessary, because reciprocation provides a way to cooperate without becoming vulnerable to exploitation. This means that players must interact multiple times.
4. The environment should have minimal constraints on interactions between agents, so that methods of cooperation will be suitable for a wide variety of environments.

We have tried to satisfy all these criteria by generating each single interaction of two agents from a probability distribution over a large class of normal form games where each normal form game is randomly generated and played only once. Since each game is randomly generated, it is extremely unlikely that it will be a fixed sum game or that the payoffs for each player will be equal. Cooperation in this environment is possible, and so is exploitation. Since agents play with each other multiple times, even though the game is different every time, they have the opportunity to reciprocate their opponent's cooperative or exploitative moves. The problem of determining which moves are cooperative and which are exploitative is left to the agents.

There are several desirable properties for agents in this environment:

1. They should not be vulnerable to a hostile opponent. Their payoff should not drop below the best payoff they could achieve if their opponent had the sole goal of reducing their payoff.
2. They should be able to achieve cooperation. When playing against another agent which is willing to cooperate they should be able to jointly increase their payoffs.
3. They should not be vulnerable to exploitation. They should only cooperate if their opponent is cooperating as well. Unreciprocated cooperation over the short term is reasonable (such as on the first move in Tit-for-Tat), but if the opponent has a history of not cooperating an agent should not continue to cooperate.

Our method of achieving cooperation is based on a parameter driven modification of the original game, where the parameters model the attitude of each agent towards the other agent. For any given game, we construct a modified game using the attitudes of both players and calculate its Nash equilibrium. If both players have positive attitudes and play the Nash equilibrium of the modified game, our experimental results show that their expected payoffs in the original game is higher than if they had played a Nash equilibrium of the original game. A Nash equilibrium is a pair of probability distributions over the moves of each agent. If both agents play a Nash equilibrium then neither agent will have an incentive to change from the Nash equilibrium.

This method of cooperation requires that an agent know the attitude of its opponent. Since this information is generally not available and potentially not constant (an opponent can change its attitude), we present an algorithm for an agent to estimate its opponent's attitude that allows it to chose its own attitude accordingly. Note that for a particular game there may be multiple Nash equilibria. Therefore our algorithm is also capable of learning how its opponent chooses a specific Nash equilibrium to play.

Related Work

Research in multi-agent systems has shown that when agents cooperate the social welfare increases. We use the concept of attitude to achieve cooperation. (Levine 1998; Rabin 1993; Sally 2002) describe ways of using attitude to explain the cooperation of people when playing certain types of games. They include a sympathy factor to reflect the fact that people prefer to cooperate with people who cooperate with them. We have not taken sympathy into account in our system, but we could model it by altering the prior over the opponent attitude and its beliefs about the agent attitude. Attitude is not the only way to explain cooperation. For instance, (Altman, Bercovici-Boden, & Tennenholtz 2006) present a method of predicting the behavior of human players in a game by using machine learning to examine their previous behavior when playing different games. In (Saha, Sen, & Dutta 2003) a mechanism is proposed where agents base their decisions about whether or not to cooperate on future expectations as well as past interactions.

Given a static environment, it is possible to learn how to play using fictitious play (Fudenberg & Levine 1998), but this approach does not produce cooperation, and it requires repeated exposure to a single game. A stochastic game (Shapley 1953) is a repeated set of games between players where the payoffs of each game are determined by the current state, and the outcome of each game affects the subsequent state. A number of approaches have been developed to learn stochastic games (e.g., (Shoham, Powers, & Grenager 2003)), but most focus on achieving the best individual payoff and not on cooperation, and they require that the environment consist of a limited number of states. Stochastic games represent a midpoint between repeated play of a single game and our environment, where a game is never seen twice. Algorithms using reinforcement learning for stochastic games only need to know the current state and the payoff received in the previous state. A variation of Q-learning which can achieve cooperation in self play while avoiding exploitation is described in (Crandall & Goodrich 2005). In our environment agents need to know their opponent's payoffs because they do not have the ability to observe the opponent's prior play for the current game; the opponent's payoffs are the only information agents can use to try to predict their opponent's play.

This paper does not discuss the problem of cooperation in a single interaction. To design an agent that is capable of cooperating in a single interaction it would be useful to look at focal point theory (Kraus, Rosenschein, & Fenster 2000), which enables coordination between two agents without communication.

Description

Environment

The environment we have chosen consists of repeated play of randomly generated normal form games. After exploring a number of alternatives, we have found that 16-move normal form games with payoffs drawn from a uniform distribution between 0 and 1 provide opportunities for cooperation without making cooperation the only reasonable choice. Increasing the number of moves per agent causes the environment to become too computationally expensive without changing the nature of the game. Reducing the number of moves per agent reduces opportunities for cooperation. We have explored generating payoffs from a normal distribution, but found that this also reduced the opportunities for cooperation. Running 1000 iterations allows sufficient time for agents to adjust to the play of their opponent with a high degree of accuracy. More details on the experiments we have done are in (Damer & Gini 2008).

The sequence of play in our environment is as follows:

1. Generate a game by assigning each agent 16 moves, and drawing a payoff for each agent for each combination of moves from a uniform distribution from 0 to 1.
2. Allow both agents to observe the game and simultaneously select a strategy for the game which consists of a probability distribution over possible moves.
3. Draw a move for each agent from the probability distribution that agent provided.

4. Award each agent the appropriate payoff for the pair of moves chosen.
5. Inform each agent of the move chosen by its opponent.

This is a good environment in which to study cooperation because agents' interests are neither diametrically opposed nor identical, so cooperation is possible without being mandatory. Agents in this environment must determine how to cooperate on their own without any environmental cues, and, because the games are randomly generated, they must be able to cooperate in a wide variety of situations.

Attitude and Belief

We use a modification of the original game to achieve cooperation. Each agent selects an *attitude* which reflects the degree to which it is willing to sacrifice its own score to improve its opponent's score. An attitude is a real number, in the range between -1 and 1. An attitude of 1 means that the opponent's payoff is valued as highly as the agent's own payoff. An attitude of 0 means that the agent is indifferent to the opponent's payoff. An attitude of -1 means that the agent is only concerned with how well it does in comparison to its opponent.

A modified game is created in which each agent's payoff is equal to its payoff from the original game plus its attitude times the payoff of its opponent in the original game. Given a game G with payoff functions g_{agent} and g_{opp} we construct the modified game G' as follows:

$$g'_{agent} = g_{agent} + \text{att}_{agent} * g_{opp} \qquad (1)$$

where att_{agent} indicates the attitude of the agent and g_{opp} indicates the payoff of its opponent.

When agents select their moves from the Nash equilibria of the modified game and play those moves in the original game the expected score of each agent improves. Figure 1 shows the effect of different combinations of attitudes on the payoff of a player. When both agents adopt an attitude of 1, they can improve their average payoffs from .80 to .90. Even when they only adopt an attitude of .2 their payoffs improve to .87.

So far we have assumed that agents know the attitudes of their opponents. Since this information is not generally available, an agent needs to estimate the attitude of its opponent. We call the value an agent uses for an estimate of its opponent's attitude its *belief*. We have explored alternatives such as assuming an indifferent opponent (belief is 0) or assuming a reciprocating opponent (belief is equal to agent's attitude) and found that those assumptions prevent effective cooperation. If an agent assumes that its opponent is indifferent, then it is actually worse off when its opponent adopts a positive attitude.

To compute Nash equilibria we use the Lemke-Howson algorithm as described in (McKelvey & McLennan 1996). Note that in many games there can be multiple Nash equilibria. If agents play different Nash equilibria, or make false assumptions about their opponent's attitude, then they do not achieve cooperation. Fortunately, we will show later that an agent can learn what attitude and method of selecting Nash equilibria is used by its opponent.
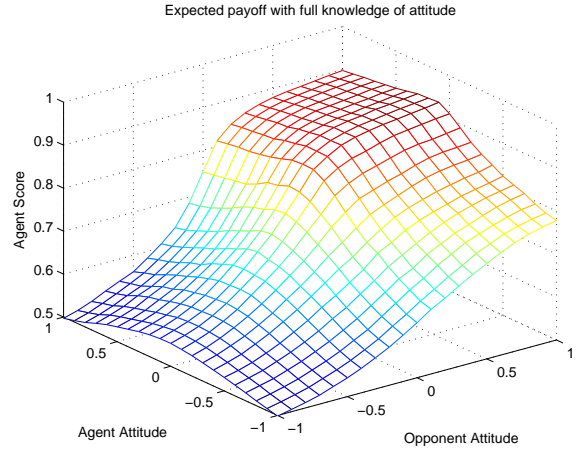

Figure 1: Payoff of an agent is affected by the attitudes of the agent and its opponent. These are aggregated results over 1000 games with 16 moves per player and with payoffs drawn from a uniform distribution between 0 and 1. The results for a single game may be quite different.

Achieving Cooperation

Once an agent has arrived at an estimate of the attitude and belief of its opponent, the question arises as to how it should use that knowledge. A self-interested approach would be to simply play the best response to the predicted strategy of the opponent. However, just as in Prisoner's Dilemma, a purely self-interested approach will not result in the best payoffs. To achieve cooperation while still avoiding exploitation, our agent sets its own attitude equal to a reciprocation level plus the attitude it has estimated for its opponent, with a maximum value of 1. The reciprocation can be quite low and still produce cooperation. We have used a reciprocation level of .1. If the opponent is not cooperative this will not lead to a significant loss for the agent, but if the opponent reciprocates in a similar way this will lead to full cooperation.

Learning Attitude and Belief

Since an agent cannot be trusted to honestly disclose its own attitude, it is necessary to learn its attitude over repeated interactions. Since an opponent's behavior is strongly influenced by its belief about the agent's attitude, it is also necessary to learn the opponent's belief. This is difficult because the only evidence available is the sequence of moves chosen in previous games. In addition, an agent can change its attitude and belief, perhaps in response to its perceptions of its opponent's attitude and belief.

The approach we propose uses Monte Carlo methods to represent a probability distribution over values of attitude and belief and methods of selecting a Nash equilibrium. We model a probability distribution over attitude and belief values using a set of particles $(p_1 \ldots p_n)$, each of which has a value for attitude p_i^{att}, belief p_i^{bel}, and a method of choosing a Nash equilibrium p_i^{nash}. Each particle's combination of attitude and belief is used to create a modified game from

an observed game, then its method of choosing a Nash equilibrium is used to find a Nash equilibrium for the modified game, which is then used to assign a probability to each move of the game. Upon observing the move chosen by the opponent, each particle is assigned a weight equal to the probability it assigned to that move. Then the set of particles is resampled with probability proportional to the weights assigned. This procedure is a variation on a particle filter (Arulampalam *et al.* 2002).

Since resampling would otherwise lead to a concentration of all the probability mass into a single particle, each of the re-sampled particles is then perturbed by a small amount. Particles are perturbed by adding a small amount of gaussian noise to p_i^{att} and p_i^{bel}. The variance of the noise is set to 10% of the error in the current estimate. Error is defined as the Euclidean distance between the true attitude and belief of the opponent and the estimated attitude and belief:

$$\text{err} = \sqrt{(\text{att}_{true} - \text{att}_{est})^2 + (\text{bel}_{true} - \text{bel}_{est})^2} \quad (2)$$

An agent does not have access to its true error, but the error can be estimated by observing the probability assigned to the opponent's move using the agent's current estimate of attitude and belief.

Our agent estimates the error in its current estimate of attitude and belief by tracking a probability distribution over fixed error levels $\{e_l | l \in 1..m\}$. The estimated error is the sum of the error levels weighted by their probability. The intuition behind this approach is that an accurate estimate will make more accurate predictions of the opponent's move, so when the opponent's actual choice of move is revealed, an accurate estimate is more likely to have predicted that move with a high probability. This allows us to use the probability assigned to the opponent's move as the basis for finding the error in the current estimate. Upon observing a move that was predicted with a particular probability, the probability of each error level is updated based on the probability of observing a move predicted with that probability given that error level and the current estimated level of cooperation.

Cooperation is defined as the correlation between an agents payoff and its opponent's payoff in the modified game which is given by:

$$\text{coop} = \frac{\text{att} + \text{bel}}{\sqrt{\text{att}^2 + 1}\sqrt{\text{bel}^2 + 1}} \quad (3)$$

It is necessary to take cooperation into account because when the level of cooperation is low, the value of being unpredictable is higher, so agents play distributions over larger numbers of moves. Figure 2 shows how changes in the level of cooperation affect the average estimated probability of the observed move. Since our method of error estimation is dependent on the estimated probability of the observed move, it is clear that it is necessary to take the level of cooperation into account.

We chose a set of error levels and divided the range of possible cooperation values $[-1, 1]$ and predicted probability values $[0, 1]$ into discrete buckets. We populated a lookup table by creating a large number of games, true attitude/belief pairs, and estimated attitude/belief pairs, and observing the
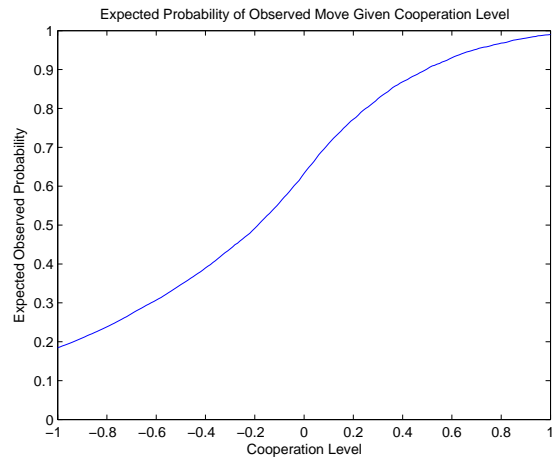

Figure 2: This graph shows how varying the level of cooperation affects the agent's choices of probability distributions over moves. With a high level of cooperation agents are more likely to assign a high probability to a few moves. With a low level of cooperation agents are more likely to assign a lower probability to many moves. This reflects the fact that unpredictability is more valuable when the opponent is not cooperating.

frequency with which moves with various predicted probabilities were observed. The lookup table has proven to be a reasonable method of approximating the probability of observing a move with a given predicted probability, given an estimated cooperation level and an error level.

Cooperative Agent Algorithm

1. Initialize
   (a) Select values for parameters
      i. n = Number of particles
      ii. r = Reciprocation level
      iii. f_{ab} = Perturbation factor for attitude and belief
      iv. f_{nash} = Perturbation factor for methods of picking Nash equilibria
      v. $\{e_l | l \in 1..m\}$ = Set of error levels
      vi. $P(e_l)$ = Distribution over error levels
   (b) Generate error estimation lookup table T with $t(j, k, l)$ equal to the probability of observing a move with estimated probability j given estimated cooperation level k and error level l, where j is a discretization of the probability, k is a discretization of the cooperation level, and l is an index into the set of error levels.
   (c) Generate initial particle set $\{p_i | i \in 1..n\}$, with attitude p_i^{att} and belief p_i^{bel} drawn from a normal distribution with mean 0 and variance 1, and method of choosing a Nash equilibrium p_i^{nash} drawn from a uniform distribution over the set of possible starting parameters of the Lemke-Howson algorithm.
2. Observe game G
3. Pick Move
   (a) Estimate opponent's parameters

i. Estimate attitude of opponent $\text{att}_{opp} = \frac{1}{n}\sum_i p_i^{att}$
ii. Estimate belief of opponent $\text{bel}_{opp} = \frac{1}{n}\sum_i p_i^{bel}$
iii. Estimate opponent's method of picking Nash equilibrium nash_{opp} from the most frequent value of p_i^{nash}

(b) Set attitude $\text{att}_{agent} = \text{att}_{opp} + r$
(c) Construct modified game G' using equation 1 and calculate its Nash equilibrium ne using nash_{opp}. ne contains two probability distributions over moves ne_{agent} and ne_{opp} which describe the mixed strategies adopted by the agent and its opponent in that Nash equilibrium.
(d) Draw move from ne_{agent}

4. Observe opponent move m
5. Update Model

(a) Update error estimate
    i. Set attitude $\text{att}_{agent} = \text{bel}_{opp}$
    ii. Construct modified game G'' and find its Nash equilibrium ne using nash_{opp}
    iii. Set $j = ne_{opp}^m$, the probability assigned by ne_{opp} to the move chosen by the opponent
    iv. Calculate cooperation value k of estimated attitude and belief using equation 3
    v. Update the probability of each error level l
       $P(e_l) = P(e_l) * t(j, k, l)$
    vi. Normalize the distribution over error levels
    vii. Estimate current level of error
        $err = \sum_{l=1}^{m} e_l * P(e_l)$

(b) Resample particles
    i. Calculate the weight for each particle
    A. Create modified game G' using p_i^{att} and p_i^{bel} and calculate its Nash equilibrium ne using p_i^{nash}
    B. Set weight for particle p_i to ne_{opp}^m
    ii. Draw n particles from the current set of particles using the calculated weights

(c) Perturb particles
    i. Modify attitude of each particle
       $p_i^{att} \sim N(p_i^{att}, err * f_{ab})$
    ii. Modify belief of each particle
        $p_i^{bel} \sim N(p_i^{bel}, err * f_{ab})$
    iii. With probability $err * f_{nash}$ draw a new method of calculating Nash equilibria for each particle.

Evaluation

Figure 3 shows the speed at which our algorithm can learn the attitude and belief used by a stationary agent. The agent's attitude and belief are randomly drawn from a Gaussian distribution with mean 0 and standard deviation 1. To choose a Nash equilibrium the agent uses a set of starting parameters drawn from a uniform distribution over all values. The error level drops fairly rapidly, but tapers off as it approaches zero. This is because less disconfirming evidence is seen as the error level drops, so there are fewer opportunities to learn. Note that 100% predictive accuracy is not achieved despite a low level of error. This is because agents that are not fully cooperative tend to use randomization when picking their moves.

Figure 4 shows the speed at which our algorithm can achieve cooperation in self play. Both agents are simulta-
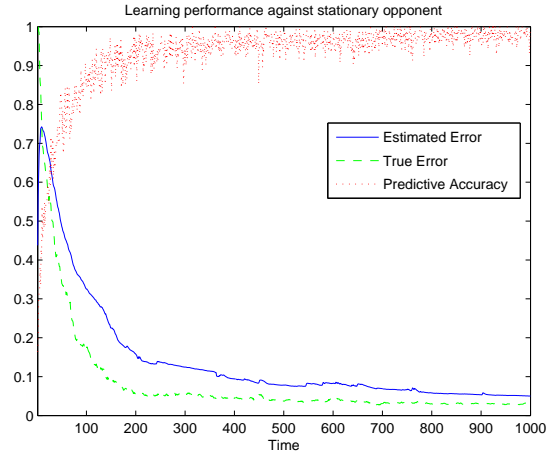

Figure 3: Efficiency with which an agent can learn a static attitude and belief. The dotted line shows the accuracy of the prediction in terms of the ratio between the estimated probability of the opponent choosing its move and the actual probability the opponent assigned to its move. Results are aggregated over 100 runs. Agents had also to learn their opponent's choice of Nash equilibrium.
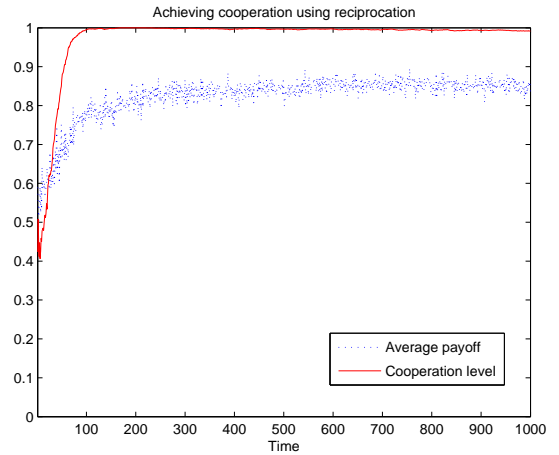

Figure 4: Speed at which cooperation can be achieved in self play. The solid line shows the level of cooperation, and the dotted line shows the payoff achieved by the agents. Results are aggregated over 100 runs. Agents had also to learn their opponent's choice of Nash equilibrium.

neously learning the attitude and belief of the other agent, and then setting their own attitude equal to .1 greater than the other agent's attitude, with the constraint that their own attitude must fall between 0 and 1.

Analysis of Proposed Approach

A limitation to the particle filter based approach to predicting an opponent's behavior is that it is only capable of learning a limited set of opponent strategies. It would be best if

an agent could learn the policy that its opponent is using to determine a distribution over moves given the payoff matrix for the game from the space of all possible policies. However, the space of all possible policies is $\triangle_n^{R^{2n^2}}$; the set of all functions from games (R^{2n^2}) onto probability distributions over moves ($\triangle_n$ - the n-dimensional simplex), where n is the number of moves for each player. This space is too large to learn efficiently, but it also contains many irrational policies which can be eliminated from consideration. For example, it includes the policy of always playing move 1 regardless of the payoffs. The set of policies considered by our algorithm is much smaller than the set of all possible policies, but it includes cooperative policies as well as policies which can avoid exploitation and defend against a hostile opponent. Furthermore, the particle filter model is easy to extend with other policies as they arise.

One important question in our approach is how to determine the level of perturbation applied to particles after re-sampling. If the particles are perturbed too much, the system will never converge to an accurate estimate due to the level of noise introduced by the perturbation. On the other hand, if the particles are not perturbed enough then the system will converge extremely slowly because the level of perturbation will not be sufficient to prevent all the probability mass from being assigned to a single particle. Our system sets the level of perturbation to a constant times the estimate of the current level of error. There is no theoretical basis for this, but in practice it has proven successful.

Because many games have multiple Nash equilibria our algorithm must learn which method of selecting a Nash equilibrium is used by the opponent. This is another very difficult problem, since the number of different ways of selecting a Nash equilibrium is limited only by human ingenuity. We have demonstrated our algorithm using a set of arbitrarily chosen methods of finding Nash equilibria. We use a deterministic algorithm to find Nash equilibria which can return different equilibria depending on its starting parameters. We use different values of the starting parameters to simulate the problem of learning how the opponent selects a Nash equilibrium. The algorithm could easily be extended by including other methods of selecting a unique Nash equilibrium with whatever properties are desired.

Conclusions and Future Work

This paper describes a new environment to explore cooperation among self-interested agents. It presents an approach which can achieve cooperation in that environment, resists exploitation, and adjusts to changing attitudes.

This particle-filter-based approach to learning opponent strategies is easy to adapt to include any particular strategy, but will fail to learn any strategy that it does not consider. The size of the space of possible strategies and the large proportion of irrational strategies in that space suggest that it is not useful to attempt to include every strategy.

The current algorithm requires prior knowledge of the distribution from which games are drawn. A generalization is to develop an error estimator that does not require that prior knowledge.

References

Altman, A.; Bercovici-Boden, A.; and Tennenholtz, M. 2006. Learning in one-shot strategic form games. In *Proc. European Conf. on Machine Learning*, 6–17. Springer.

Arulampalam, M. S.; Maskell, S.; Gordon, N.; and Clapp, T. 2002. A tutorial on particle filters for online nonlinear/non-gaussian Bayesian tracking. *IEEE Transactions on Signal Processing* 50(2):174–188.

Axelrod, R. M. 1984. *The evolution of cooperation*. Basic Books.

Crandall, J. W., and Goodrich, M. A. 2005. Learning to compete, compromise, and cooperate in repeated general-sum games. In *Proc. of the Int'l Conf. on Machine Learning*, 161–168. New York, NY, USA: ACM.

Damer, S., and Gini, M. 2008. A minimally constrained environment for the study of cooperation. Technical Report 08-013, University of Minnesota, Department of CSE

Fudenberg, D., and Levine, D. K. 1998. *The Theory of Learning in Games*. MIT Press.

Govindan, S., and Wilson, R. 2008. Refinements of Nash equilibrium. In Durlauf, S. N., and Blume, L. E., eds., *The New Palgrave Dictionary of Economics, 2nd Edition*. Palgrave Macmillan.

Kraus, S.; Rosenschein, J. S.; and Fenster, M. 2000. Exploiting focal points among alternative solutions: Two approaches. *Annals of Mathematics and Artificial Intelligence* 28(1-4):187–258.

Levine, D. K. 1998. Modeling altruism and spitefulness in experiments. *Review of Economic Dynamics* 1:593–622.

McKelvey, R., and McLennan, A. 1996. Computation of equilibria in finite games. In *Handbook of Computational Economics*.

Nowak, M. A., and Sigmund, K. 1993. A strategy of win-stay, lose-shift that outperforms Tit for Tat in the Prisoner's Dilemma game. *Nature* 364:56–58.

Rabin, M. 1993. Incorporating fairness into game theory and economics. *The American Economic Review* 83(5):1281–1302.

Saha, S.; Sen, S.; and Dutta, P. S. 2003. Helping based on future expectations. In *Proc. of the Second Int'l Conf. on Autonomous Agents and Multi-Agent Systems*, 289–296.

Sally, D. F. 2002. Two economic applications of sympathy. *The Journal of Law, Economics, and Organization* 18:455–487.

Shapley, L. S. 1953. Stochastic games. *Proceedings of the NAS* 39:1095–1100.

Shoham, Y.; Powers, R.; and Grenager, T. 2003. Multi-agent reinforcement learning: a critical survey. Technical Report, Stanford University.

Trivers, T. 1971. The evolution of reciprocal altruism. *Quarterly Review of Biology* 36:35–57.