Learning To Cooperate in a Social Dilemma: A Satisficing Approach to Bargaining

Jeffrey L. Stimpson & Michael A. Goodrich JSTIM,MIKE@CS.BYU.EDU
Computer Science Department, Brigham Young University, Provo, UT 84602

Abstract

Learning in many multi-agent settings is inherently *repeated play*. This calls into question the naive application of single play Nash equilibria in multi-agent learning and suggests, instead, the application of give-and-take principles of bargaining. We modify and analyze a satisficing algorithm based on (Karandikar et al., 1998) that is compatible with the bargaining perspective. This algorithm is a form of relaxation search that converges to a satisficing equilibrium without knowledge of game payoffs or other agents' actions. We then develop an M action, N player social dilemma that encodes the key elements of the Prisoner's Dilemma. This game is instructive because it characterizes social dilemmas with more than two agents and more than two choices. We show how several different multi-agent learning algorithms behave in this social dilemma, and demonstrate that the satisficing algorithm converges, with high probability, to a Pareto efficient solution in self play and to the single play Nash equilibrium against selfish agents. Finally, we present theoretical results that characterize the behavior of the algorithm.

1. Introduction

Many multi-agent learning problems can be viewed as social dilemmas. For example, in (Goodrich et al., 2003) we presented a multi-robot scenario that illustrated the difficulties in creating learning algorithms for environments where there are multiple learning agents and where games are non-zero sum. These difficulties arose because each robot needed to use a common resource, but if each robot tried to dominate the resource then every robot suffered. This is typical of prisoner's dilemma-like environments with ongoing interactions; the robots were required to act independently, but the solution concept of a single-play Nash equilibrium was inappropriate for these repeated interactions.

In this paper, we introduce a multi-agent social dilemma game that has the essential characteristics of the prisoner's dilemma, but is broad enough to represent social dilemmas with more than two actions and more than two agents. We then present a satisficing algorithm that allows agents to learn a satisficing equilibrium to the game. Simulation results are presented that compare the satisficing algorithm to Q-learning and a general form of belief-based learning. We evaluate the performance of each algorithm from a bargaining perspective and determine (I) whether the algorithm reaches a Pareto efficient solution in self-play, and (II) whether the algorithm avoids exploitation by selfish agents. We demonstrate that the satisficing algorithm satisfies both properties, even when the agents do not know the game structure or the possible actions of other agents. We conclude by evaluating some of the theoretical properties of the satisficing algorithm.

2. Related Literature

The literature in multi-agent choice is vast and space is limited, so we cite only a few. A more complete citation list can be found in (Stimpson, 2002). Machine learning researchers have explored many approaches to learning in games. (Hu & Wellman, 1998; Claus & Boutilier, 1997) presented extensions to Q-learning for stochastic games that converge to Nash equilibrium solutions, and (Bowling & Veloso, 2000) has extended one of these algorithms to exploit the naive strategies of other agents. Complementing these papers is work from the economics literature (Kalai & Lehrer, 1993; Fudenberg & Levine, 1998) that describes when and how model-based agents tend to converge to a Nash equilibrium. Others (Mundhe & Sen, 2000; Hu & Wellman, 2000), have explored how reinforcement learning algorithms proceed when assumptions required for convergence to a Nash equilibrium are violated.

Unfortunately, a lesson learned from the repeated play Prisoner's Dilemma game (Axelrod, 1984) is that strategies that tend to Nash equilibria are not always desirable when agents engage in repeated interactions. Attempts to generate cooperative solutions using algorithms with claims of bounded rationality have offered some insight into when cooperation is preferred to Nash equilibria (Mor & Rosenschein, 1995). From a machine learning perspective, augmenting state information with coordination-specific information can lead to cooperation (Boutilier, 1999).

Literature on multiple-player or multiple-action social dilemmas is limited. One formal discussion is provided in (Hamburger, 1973). (Luce & Raiffa, 1957) and (Bendor & Mookherjee, 1987) briefly discuss a multiple-player, two-action prisoner's dilemma. In addition to multiple players, the prisoner's dilemma has been extended to continuous degrees of cooperation (Frolich & Oppenheimer, 1996).

3. A Social Dilemma

In this section, we introduce the *multi-agent social dilemma* (MASD) which is a game with the same essential characteristics as the prisoner's dilemma, but which allows for multiple players and actions. This game is useful for illustrating strengths and weaknesses of various multi-agent learning algorithms.

Consider a system consisting of N agents. At each iteration, every agent is faced with a decision of allocating M units of some discrete resource towards two possible goals S_i and G. S_i is some purely self-interested goal for agent $i \in \{1, ..., N\}$ and G is some group goal for all agents. Let u_i be the amount contributed by agent i towards the group goal G (and thus $M - u_i$ is the amount contributed to the selfish goal S_i). Let $\mathbf{u} = [u_1, \ldots, u_N]$ denote the vector of all actions taken by the agents. For each agent there are $M + 1$ possible values for $u_i \in \{0, 1, 2, ...M\}$. Let each agent's total utility be represented as a linear combination of the total amount contributed to the group goal G and the amount individually contributed to his or her own selfish goal S_i. The utility to agent i given the actions of all agents is

$$R_i(\mathbf{u}) = k_{Gi}\Big[\sum_{j=1}^{N} u_j\Big] + k_{Si}(M - u_i), \qquad (1)$$

where k_{Si} is agent i's weighting of his or her own selfish goal and k_{Gi} is agent i's weighting of the group goal.

Suppose that all agents have the same $k_S = k_{S_i}$ and $k_G = k_{G_i}$the relative. Assuming that relative (not absolute) utilities are important, we can reduce the number of parameters by letting $k_G = \frac{1}{NM}$ and $k_S = \frac{k}{M}$ where k is a positive constant. When $k < 1$ it means that each agent values a unit of contribution towards the selfish goal more than a unit of contribution to the group goal, and when $k > \frac{1}{N}$ it means that there is a higher potential benefit from the group goal as long as enough agents contribute to the group. Thus, attention is restricted to the case where $1 > k > \frac{1}{N}$. Substituting this reparameterization into (1), dividing by $M(1 - k)$, and dropping a constant term from the end, gives $R_i(\mathbf{u}) = \frac{[\frac{1}{N}\sum_{j=1}^{N} u_j] - ku_i}{M(1-k)}$.

It will often be useful to examine the situation from the perspective of a single agent. For these circumstances, we define $\mathbf{u_{-i}} \in \mathbf{U_{-i}}$ as the joint action of agent i's opponents. In the MASD, $\mathbf{u}_{-i}$ can be reduced to a scalar integer because the reward function depends only on the sum of the actions of agent i's opponents whence $u_{-i} = \sum_{j=1, j\neq i}^{N} u_j$, whence

$$R_i(\mathbf{u}) = R_i(u_i, u_{-i}) = \frac{(1 - kN)u_i + u_{-i}}{NM(1 - k)}. \qquad (2)$$

The most important properties of this game are summarized below. These properties illustrate the social dilemma characteristics of the game.

1. **Extreme Individual Rewards** The individual reward for agent i, R_i, is maximized when $u_i = 0$ and $u_{-i} = M(N - 1)$, and is minimized when $u_i = M$ and $u_{-i} = 0$.
2. **Extreme Average Rewards** The average reward, $\overline{R}(\mathbf{u}) = \frac{1}{NM}\sum_{i=1}^{N} u_i$, is maximized at $\overline{R} = 1$ by the joint action $\mathbf{u}$ where $\forall i\ u_i = M$, and is minimized at $\overline{R} = 0$ when $\forall i\ u_i = 0$
3. **Nash Equilibrium** The joint action $\mathbf{u}$ where $\forall i\ u_i = 0$ is both strategically dominant and the unique Nash equilibrium.
4. **Nash Bargaining Solution** When the fallback position is defined as the strategically dominant solution, the joint action $\mathbf{u}$ where $\forall i\ u_i = M$ is the Nash Bargaining solution. It is therefore also Pareto optimal.

This final property is a key for discriminating between various learning algorithms. We will demonstrate that many algorithms fail to discover a Pareto efficient solution in self play. The satisficing algorithm, by contrast, discovers a Pareto efficient solution with high probability given a wide range of initial parameters, and tends toward the Nash Bargaining solution in self-play.

4. The Satisficing Algorithm

Herbert Simon introduced the term satisficing to mean "good enough" (Simon, 1996). Although he discussed satisficing from several perspectives, a frequent perspective was one in which an agent searched through a set of possible decisions until a decision was found which had utility that exceeded an aspiration level. A formal treatment of this algorithm was analyzed in a prisoner's dilemma context in (Karandikar et al., 1998) and further analyzed in (Stimpson et al., 2001) for deterministic updates. The conclusion of these papers is that a satisficing algorithm can lead to mutual cooperation in the prisoner's dilemma under a broad variety of conditions.

4.1 Extending Karandikar's Algorithm to the MASD

(Karandikar et al., 1998)'s algorithm works as follows: (a) when the aspiration level, α, is not met, the agent switches actions, and (b) the aspiration level is updated as the convex combination of the old aspiration and the current reward via learning rate λ. In the prisoner's dilemma, switching means simply switching to the other action. In the social dilemma, an agent must choose between an arbitrary number of actions. We adopt the simple method of selecting the next action randomly; more sophisticated techniques, such as policy hill climbing, are topics for future work. Figure 1 states the modified satisficing algorithm in the MASD context. For simplicity, we suppose that all agents use the same learning rate λ.

4.2 An Example of Satisficing Learning in the MASD

Figure 2 illustrates the satisficing learning process for two agents. The figure is shown for $M = 10$ and initial aspirations are $(\alpha_1, \alpha_2) = (1.5, 2.0)$. In the figure, each open circle denotes a possible reward for some joint action, where the x-coordinate is Player 1's reward and the y-coordinate is Player 2's reward. At each time step, one of these rewards is determined from the joint action. The line that trails down from the upper right corner of the graph is a plot of the aspiration history for the agents. The gray area to the northeast of the aspiration level is termed the *satisficing region*, meaning that if a reward is selected that is in this region, both players will be satisfied and aspirations will converge to the chosen reward.

At each iteration t

1. For each agent, compute
$$R_i(\mathbf{u}(t)) = \frac{[\frac{1}{N}\sum_{j=1}^{N} u_j(t)] - k u_i(t)}{M(1-k)}$$
2. Update the actions for satisficing agents
   - If $R_i(\mathbf{u}(t)) \geq \alpha_i(t)$ then $u_i(t+1) = u_i(t)$ otherwise select $u_i(t+1)$ from a uniform distribution over all actions.
3. Update the aspirations for satisficing agents
   - $\alpha_i(t+1) = \lambda\alpha_i(t) + (1-\lambda)R_i(\mathbf{u}(t))$

Figure 1. The satisficing algorithm for the MASD.

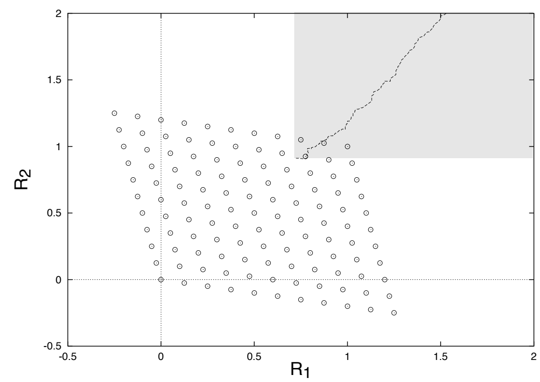

Figure 2. An illustration of the satisficing learning process. For this example, $M = 10$, $k = 0.6$, and $\lambda = 0.99$ for both agents.

Initially, all actions produce rewards that are less than the aspiration levels of the agents. This causes aspirations to drop and, as a result, the agents are choosing randomly and thus the rewards are also randomly selected from any of the possibilities shown.

At the time shown in the figure, the satisficing region intersects the area of feasible rewards. It is now possible that a single agent may be satisfied with an action. However, because the aspirations are still quite high, most of those individually satisficing actions are likely to exploit the other agent. An agent is therefore unlikely to stay satisfied for more than a few iterations because the unsatisfied agent will constantly be changing. During this time, if both agents chose $M = 10$ then they both receive a reward of (1,1) and they both continue to play this satisficing action ever after. Once this action is chosen, the aspiration vector approaches (1,1) until eventually the only action that is mutually satisficing is mutual cooperation.

This is the typical manner in which satisficing converges in the two-player, multiple-action MASD. Intuitively, we can see that mutual cooperation is the most probable outcome as long as aspirations start high and aspirations are updated slowly. In most cases, $\mathbf{u} = (M, M)$ will be the first joint action that is satisficing to both agents.

5. Simulation Results

In this section, we analyze expected average reward in self-play for (a) a general belief-based learner, (b) the Q-learning algorithm, and (c) the satisficing algorithm. Note that we do not present simulation results from other state-of-the-art algorithms such as the WoLF PHC algorithm (Bowling & Veloso, 2000) because these algorithms purport to settle on single play Nash equilibrium solutions, and these solutions may be inappropriate for the multi-agent social dilemma. Furthermore, simple strategies such as random play or play fixed at a select action, are either subject to exploitation by other agents, or are needlessly pessimistic.

To compare the algorithms, it is useful to have a standard of comparison. In related work, Bowling and Veloso (Bowling & Veloso, 2000) state that a good learning algorithm should reach a Nash equilibrium in self-play and should find a best-response against inferior opponents. We flip these desiderata to identify two properties that are desirable from the bargaining perspective: a good learning algorithm should (I) reach a Pareto efficient solution in self-play and (II) should not be exploited by selfish agents. We will use average performance in self-play as a metric for measuring how often Pareto efficient solutions are obtained.

5.1 Belief-Based Learning

In this section, we present and discuss a general form of belief-based learning described in (Feltovich, 2000). In this algorithm, player's beliefs about an opponent's play are characterized by a set of weights for each opponent action. At time t player i creates a probabilistic model, $q_i(u_{-i}; t)$ of all other agents using standard techniques from fictitious play (Fudenberg & Levine, 1998). Given this opponent model, a player can compute the expected value, $\hat{V}_i(u_i; t)$, for each action u_i as $\hat{V}_i(u_i; t) = \sum_{u_{-i} \in U_{-i}} R_i(u_i, u_{-i}) q_i(u_{-i}; t)$. A probability, $p_i(u_i; t)$, of choosing action u_i is then assigned as follows (thereby producing mixed strategies),

$$p_i(u_i; t) = \frac{\exp(\lambda_i \hat{V}_i(u_i; t))}{\sum_{u'_i \in U_i} \exp(\lambda_i \hat{V}_i(u'_i; t))},$$

where λ_i is the Boltzmann parameter that determines how optimally player i plays according to his beliefs. Note that this algorithm is a general case of many well-known belief-based learning algorithms including standard and cautious fictitious play (Fudenburg & Levine, 1998).

Consider this learning model applied to the MASD. Substituting Equation (2) for R_i into the probability of choice and reducing leads to

$$p_i(u_i; t) = \frac{e^{-A\lambda_i u_i}}{\sum_{z=0}^{M} e^{-A\lambda_i z}}. \tag{3}$$

where $A = \frac{1-kN}{NM(1-k)}$. Note that these probabilities are completely independent of the opponent's strategies or the player's predictions about the probabilities of the opponents' play. This means that learning models of this form are unable to adapt their behavior to their opponents in the MASD, and essentially reduce to a purely random strategy with the above distribution function. Furthermore, it can be shown that any dependence on state (whether from game history or player history) is eliminated in the final probability distribution.

Consider the expected play for two extreme values of λ_i. When $\lambda_i = 0$, the expected play is $\frac{M}{2}$, and in the limit as $\lambda_i \to \infty$, then the expected play is 0. When all agents in a society use a learning strategy of this type, $\overline{R}$ is bounded in $[0, \frac{M}{2}]$ depending on the values for λ_i. Experiments were conducted for various parameter values where, in a given game, all agents used the same λ. When $N = 5$, $M = 3$, and $k = 0.6$, the theoretical payoffs $\overline{R}$ are as follows: $\lambda = 0 \to \overline{R} = 0.5$, $\lambda = 1 \to \overline{R} = 0.36$, and $\lambda = 10 \to \overline{R} = 0.012$. Note that since these values are supported with the average empirical results, plots of the simulations are omitted.

In terms of the two desiderata, since the belief based agents act randomly, they will not learn mutual cooperation. They can, however, avoid exploitation by an appropriate choice of parameters.

5.2 Q-Learning

In strict terms, applying Q-learning to multiagent environments is not mathematically justified due to the fact that the transition function is not stationary when the other agents are able to learn and adapt their behavior. Such limitations are addressed in algorithms that adopt a stochastic games framework (such as WoLF (Bowling & Veloso, 2000)), but these algorithms emphasize convergence to single play Nash equilibria. Because of the non-stationarity, Q-learning has been shown to sometimes converge to Pareto efficient solutions in the prisoners' dilemma (Sandholm & Crites, 1995), and this is why it is instructive to study Q-learning in the MASD.

We designed several experiments to evaluate the performance of Q-learners in the MASD. The main results are presented in Figure 3, which displays the average rewards $\overline{R}$ throughout the learning process for three different systems of Q-learning agents. As can be seen,

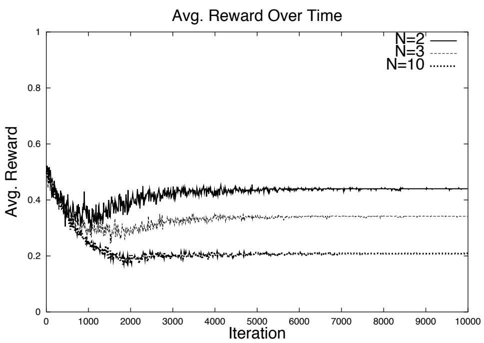

Figure 3. The average reward of Q-learning agents over time in the MASD. The three lines represent three separate experiments with N=2, N=3, and N=10. In all cases, $M = 1$ meaning that each agent has two choices. Each experiment consisted of averaging the rewards of all the agents over 200 trials. The game parameter k was chosen from a uniform random distribution over its legal range given N. The Q-learners used a fixed learning rate $\alpha = 0.2$, a discount factor $\gamma = 0.9$ and Softmax exploration.

in all cases, cooperation was relatively infrequent, although we found that the Q-learners always converged. In most cases, agents converged to the Nash equilibrium, but occasionally mutual cooperation emerged.

We varied both the parameters of the agents (α, γ, state representation) and the properties of the game (N, M, and k). Except for N, the performance of the Q-learning agents is not highly dependent on the agent parameters. For example, there is a wide range of values for both α, γ, and M that lead to similar results. We experimented with different state representations as well (account for the previous entire joint action, $\mathbf{u}$, and account for the sum of the joint action, u_{-i}) but found that it did not have a significant effect on the frequency of cooperation.

In terms of the two desiderata, the Q-learners tend to learn best responses to stationary strategies (as evidenced by the predominance of Nash equilibrium solutions), so they are unlikely to be exploited. However, they only rarely learn mutual cooperation.

5.3 The Satisficing Algorithm

Figure 4 displays the average reward produced by the satisficing algorithm for two agents as a function of M in self-play. First, note that the performance is very high, meaning that in self-play the satisficing algorithm is likely to converge to a Pareto efficient solution. Note also that as M increases, the average reward decreases, but stays fairly high, even though the probability of guaranteed cooperation gets very small which means that the algorithm degrades gracefully as complexity increases. This can be accounted for by considering that mutual cooperation can still occur even when we cannot guarantee it. Also, the primary reason that $\mathbf{u^c}$ becomes more difficult to obtain is not that bad solutions are found, but that fairly good solutions are found that are close to mutual cooperation.

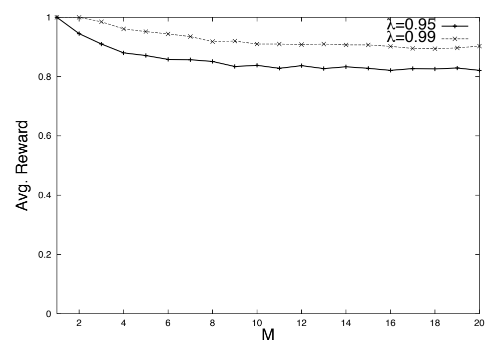

Figure 4. The average reward for 500 games over the society of satisficing agents. In all games, $k = 0.6$ and initial aspirations were randomly selected from the range [1.5,2.0].

Figure 5 compares the average reward over the society as N increases for three different values of M. As M increases, there are more solutions that are not Pareto efficient and it is therefore more difficult to find a Pareto efficient solution. We note that $\overline{R}$ starts high, but falls of as N increases. By the time $N = 10$, for moderate values of M, $\overline{R}$ has significantly decreased and begins to approach 0.5.

In terms of the two desiderata, mutual cooperation is likely to emerge in self-play. Furthermore, we can prove that the algorithm is likely to converge to a single play Nash equilibrium when playing against a society of selfish agents, which means that the algorithm is not likely to be exploited. A corresponding theorem exists that states conditions that guarantee the algorithm will converge, with high probability, to mutual cooperation in self-play. In the interest of space, we present only the first theorem in this paper.

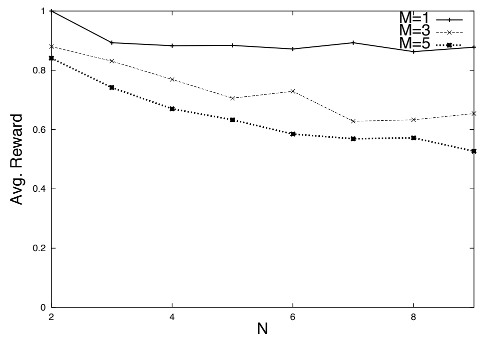

Figure 5. The average reward for 500 games of a society of satisficing agents as N increases. k was selected randomly from its legal range. Initial aspirations were chosen randomly between $R_{\max}$ and $2R_{\max}$. All agents used a $\lambda = 0.99$.

6. Satisficing Against Defecting Agents

If a learning agent is facing agents that always attempt to exploit others, an effective learning algorithm should be able to learn the single play Nash equilibrium. In this section, we evaluate the ability of a satisficing agent to learn $u_i = 0$ in such a society. Observe that u_{-i} will always be 0 for the satisficing agent. This means that the reward to the satisficing agent i for taking action u_i is

$$R_i(u_i(t), u_{-i}(t) = 0) = \frac{1 - kN}{NM(1-k)} u_i(t). \quad (4)$$

Note that $R_i \leq 0$ which implies that $\alpha(t)$ will always be decreasing until $\alpha(t) \leq 0$. At that point, whenever $R_i \geq \alpha(t)$, the agent will be satisfied indefinitely. As long as $0 < \lambda < 1$, the aspiration level $\alpha(t)$ cannot fall below the minimum reward given and thus the algorithm will always converge to some action u^*.

Two points need to be made before continuing. First, since we have altered Karandikar's algorithm and since the game is multi-agent, the proof of convergence in (Karandikar et al., 1998) does not apply. Second, showing that the agent converges to a single play Nash equilibrium against a selfish agent lends credibility to the argument that the satisficing algorithm will avoid exploitation, even by clever agents. The reason for this is that if a clever agent defects initially, the satisficing agent will tend toward the single play Nash equilibrium which means that the clever agent will have no incentive for changing its strategy. If the clever agent tends to cooperate initially, the satisficing agent will tend to cooperate too; if the clever agent then switches behavior toward exploitation, the satisficing agent will cease to be satisfied with cooperation and will lower its aspiration until it learns the single play equilibrium. Thus, it is unlikely that the satisficing agent will converge to a steady state solution that can be exploited. It is, however, possible for a coalition of clever agents to manipulate a satisficing agent.

6.1 Intuition Behind the Argument

The best response for a satisficing agent against $u_{-i} = 0$ is $u_i = 0$; this is the action that is most likely to be produced by the satisficing algorithm. This result requires that the initial aspiration $\alpha(0) \geq R_i(0,0) = 0$. This means that the satisficing agent will be initially unsatisfied for several iterations while aspirations fall towards zero. Eventually, at some t_0, $\alpha(t_0) < 0$, after which if $u_i(t) = 0$ is chosen for $t > t_0$ then the agent will converge to the Nash equilibrium. However, it is possible at some time t_1 for aspirations to fall below $R_i(1,0)$ before full defection is chosen. The trick is to make $T = t_1 - t_0$ large enough that $u_i = 0$ is chosen with a high probability.

Figure 6 illustrates this concept. The aspiration starts well above the reward for mutual defection denoted by $R(0)$. At each iteration, however, α falls towards the received payoffs. At some point t_0, $\alpha(t)$ drops below the reward for playing the Nash equilibrium. At this point only the Nash equilibrium is satisficing. Eventually, at some time $t_1 > t_0$, if the agent does not play $u_i = 0$, the aspirations will fall below $R_i(1,0)$, after which it is possible to converge to $u = 1$, and thus be indefinitely exploited.

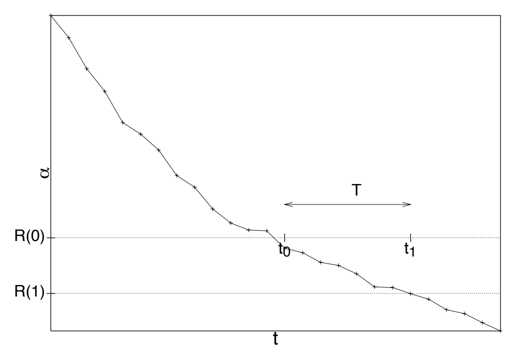

Figure 6. An example of the aspirations of a single satisficing agent against a society of defecting agents over time.

6.2 Theorem

The critical factor in determining if the algorithm will converge to the Nash equilibrium is the length of the interval $T = t_1 - t_0$. We can place a lower limit on T by identifying the value of T that causes aspirations to fall most sharply between $R_i(0,0)$ and $R_i(1,0)$. We refer the reader to (Stimpson, 2002) for a complete derivation. This bound is given by

$$T \quad \geq \quad \log_\lambda \left[\frac{M-1}{M}\right] - 1,$$

which depends only on λ and M. Note that as λ approaches one, T gets larger and approaches infinity, but as M goes to infinity, T goes to zero. We can now state and prove the following:

Lemma 6.1 *Assuming $\alpha(0) \geq R_i(0,0)$, then for any $T' > 1$, there exists a $\lambda \in (0,1)$ such that the shortest interval T in which $\alpha(t_0 + T) > R_i(1,0)$ satisfies $T > T'$.*

Proof. Since $T \geq \log_\lambda \left[\frac{M-1}{M}\right] - 1$ it suffices to find a λ such that $\log_\lambda \left[\frac{M-1}{M}\right] - 1 \geq T'$for any $T' > 1$. Such a λ must satisfy $\frac{\ln\left(\frac{M-1}{M}\right)}{\ln \lambda} \geq T' + 1$ which is equivalent to $\left(\frac{M-1}{M}\right)^{\frac{1}{T'+1}} \leq \lambda \ < \ 1$. Thus, we can always choose a $\lambda \in (0,1)$ that will make T greater than any arbitrary $T' > 1$. ∎

We now know conditions on λ such that there is a time window of at least T iterations in which the one shot Nash equilibrium action, $u_i = 0$, will be the only satisficing action. During this window, actions are selected from a uniform distribution where $P[u = 0] = \frac{1}{M+1}$. It follows, then, that the probability of the Nash equilibrium occurring in this window of length T is given by $1 - (\frac{M}{M+1})^T$. The Nash equilibrium could be reached in subsequent iterations (after $\alpha(t) < R(1,0)$), but that will only increase the probability that $u^* = 0$. Thus, the probability that the agent learns the Nash equilibrium against a society of always-defecting agents can be bounded by

$$P[u_i^* = 0] \geq 1 - \left(\frac{M}{M+1}\right)^T. \tag{5}$$

Theorem 6.1 *Consider a multiagent social dilemma specified by (N, M, k) played by a single satisficing agent i when $u_{-i} = 0$. Suppose that $\alpha(0) \geq R_i(0,0)$. Then, for any ϵ such that $0 < \epsilon < 1$, there exists a learning rate λ such that the probability of the single satisficing agent learning the Nash equilibrium is at least $1 - \epsilon$.*

Proof. By Equation (5), we know that $P[u^* = 0] \geq 1 - (\frac{M}{M+1})^T$. Thus, if we can show that $1 - \left(\frac{M}{M+1}\right)^T \geq 1 - \epsilon$then it follows that $P[u^* = 0] \geq 1 - \epsilon$. To satisfy this inequality, T must satisfy $T \geq \log_{\frac{M}{M+1}}(\epsilon)$. But by Lemma 6.1, we can always choose a T such that $T \geq T' = \log_{\frac{M}{M+1}}(\epsilon)$. Thus, for any ϵ there exists a λ such that $P[u_i^* = 0] \geq 1 - \epsilon$. ∎

Empirical results confirm that $P(u^* = 0)$ is indeed bounded by this limit. A similar proof can be used to show that a learning rate λ can be chosen such that a group of N satisficing agents will likely converge to mutual cooperation (the Nash bargaining solution) if they all begin with high and similar aspiration levels.

6.3 Graceful Degradation and Convergence Time

It is desirable for the algorithm to degrade gracefully in the presence of many possible actions (M). Consider the system at any time $t \geq t_1$. At such times, the Nash equilibrium is at least as likely to be chosen as any other mutually satisficing action. Thus, the Nash equilibrium is not only possible earlier than higher values of u_i^*, but is always at least as likely as any other u_i^* as well. Furthermore, since $R_i(u_i^*, 0)$ is proportional to the ratio $\frac{u_i^*}{M}$, as M increases the probability of missing the Nash equilibrium increases, but the cost of slightly missing $u_i^* = 0$ decreases. Empirical results confirm that the average reward for a satisficing agent against $u_{-i} = 0$ degrades gracefully. The trends are similar to those shown in Figure 4 so plots are omitted in the interest of space.

Time to converge is also a very important element of the performance of the satisficing algorithm. The order of convergence time is $\left|\frac{1}{\log \lambda}\right|$, which is obtained by taking the expected aspiration level at some time t. Thus, a high λ is required to make non-exploitation likely, but it also significantly increases convergence time.

7. Discussion

We have presented an M-action, N-agent social dilemma, and evaluated the performance of several learning algorithms in this dilemma. Q-learning rarely converges to mutual cooperation in self play, and belief-based learning generates random actions without regard to the game for this dilemma. The satisficing algorithm, by contrast, usually converges to mutual cooperation in self-play, but usually avoids being exploited by selfish agents. A key point in this discussion is the assumption that a clever algorithm should learn a Pareto efficient solution in self play rather than the single play Nash equilibrium. This assumption is based on the observation that learning is often inherently repeated play, so a give-and-take approach to adaption is more appropriate than insisting on individual optimization.

The key parameter in the success of the satisficing algorithm is choosing the learning rate, λ. High values of λ make cooperation highly probable, but it may take a very long time to reach convergence. Essentially, in choosing λ, an agent must resolve a tradeoff between solution quality and convergence speed. We have assumed in this paper that the game is played for many iterations and the agents value the future, and thus solution quality will be more important than speed.

Relaxing an agent's aspirations is one way to show respect for how an agent's choices affect other agents, even in the absence of precise knowledge of the reward structure of the game and the set of possible choices of other agents. By showing respect in this way, the satisficing algorithm assumes a bargaining perspective while avoiding being exploited by selfish agents. Rather than adopt the Nash equilibrium as the basis for finding stable solutions to multi-agent learning, we instead adopt a bargaining-based solution. Bargaining, in this problem, is relaxing the aspiration level until a satisficing action is chosen. Stirling has argued that when all agents are "satisficed" then there is no incentive for any agent to change its choice (Stirling et al., 2002). In Stirling's sense, then, the satisficing algorithm produces stable solutions in self-play.

References

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

Bendor, J., & Mookherjee, D. (1987). Institutional structure and the logic of ongoing collective action. *The American Political Science Review*, *81*, 129–154.

Boutilier, C. (1999). Sequential optimality and coordination in multiagent systems. *Proceedings of the Sixteenth International Joint Conferences on Artificial Intelligence (IJCAI-99).*.

Bowling, M., & Veloso, M. (2000). Rational learning of mixed equilibria in stochastic games. In UAI2000 Workshop entitled Beyond MDPs: Representations and Algorithms.

Claus, C., & Boutilier, C. (1997). The dynamics of reinforcement learning in cooperative multiagent systems. *Proceedings of the Fifteenth National Conference on Artificial Intelligence, AAAI98.* Madison, Wisconsin USA.

Feltovich, N. (2000). Reinforcement-based vs. belief-based learning models in experimental assymetric-information games. *Econometrica*, *68*, 605–641.

Frolich, N., & Oppenheimer, J. (1996). When is universal contribution best for the group? Characterizing optimality in the prisoner's dilemma. *Journal of Conflict Resolution*, *40*, 502–516.

Fudenberg, D., & Levine, D. K. (1998). *The theory of learning in games.* MIT Press.

Fudenburg, D., & Levine, D. (1998). *The theory of learning in games.* MIT Press.

Goodrich, M. A., Crandall, J. W., & Stimpson, J. L. (2003). Neglect tolerant teaming: Issues and dilemmas. *Proceedings of the 2003 AAAI Spring Symposium on Human Interaction with Autonomous Systems in Complex Environments.*

Hamburger, H. (1973). N-persone prisoner's dilemmas. *Journal of Mathematical Sociology*, *3*, 27–48.

Hu, J., & Wellman, M. P. (1998). Multiagent reinforcement learning: Theoretical framework and an algorithm. *Proceedings of the Fifteenth International Conference on Machine Learning* (pp. 242–250).

Hu, J., & Wellman, M. P. (2000). Experimental results on Q-learning for general-sum stochastic games. *Proceedings of the Seventeenth International Conference on Machine Learning (ICML-2000).* San Francisco: AAAI Press.

Kalai, E., & Lehrer, E. (1993). Rational learning leads to Nash equilibrium. *Econometrica*, *61*, 1019–1045.

Karandikar, R., Mookherjee, D., Ray, D., & Vega-Redondo, F. (1998). Evolving aspirations and cooperation. *Journal of Economic Theory*, *80*, 292–331.

Luce, R. D., & Raiffa, H. (1957). *Games and Decisions.* New York: John Wiley.

Mor, Y., & Rosenschein, J. S. (1995). Time and the prisoner's dilemma. *Proceedings of the First International Conference on Multi-Agent Systems* (pp. 276–282). Menlo Park, CA, USA: AAAI Press.

Mundhe, M., & Sen, S. (2000). Evaluating concurrent reinforcement learners. *Proceedings of the Fourth International Conference on Multiagent Systems* (pp. 421–422). Los Alamitos, CA: IEEE Press. Poster paper.

Sandholm, T., & Crites, R. (1995). Multiagent reinforcement learning in the iterated prisoner's dilemma. *Biosystems*, *37*, 147–166. Special Issue on the Prisoner's Dilemma.

Simon, H. A. (1996). *The sciences of the artificial.* MIT Press. 3rd edition.

Stimpson, J., Goodrich, M. A., & Walters, L. (2001). Satisficing learning and cooperation in the prisoner's dilemma. *Proceedings of IJCAI 2001.*

Stimpson, J. L. (2002). Satisficing solutions to a multiagent social dilemma. Master's thesis, Brigham Young University, Provo, UT, 84602, USA.

Stirling, W. C., Goodrich, M. A., & Packard, D. J. (2002). Satisficing equilibria: A non-classical approach to games and decisions. *Autonomous Agents and Multi-Agent Systems Journal*, *5*, 305–328.