Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 08-013

A Minimally Constrained Environment for the Study of Cooperation

Steven Damer and Maria Gini

April 24, 2008

A Minimally Constrained Environment for the Study of Cooperation

Steven Damer and Maria Gini

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. This environment provides a good model of the difficulties of cooperating in an ever changing world. 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.

1 Introduction

What is cooperation? When should you cooperate? How do you cooperate? Human beings instinctively understand what cooperation is. After all, we've been instructed to 'be nice!' or 'get along!' ever since we were small children. Unfortunately there is no implementation of the BE_NICE operator for computers, so it is difficult to make a software agent which can cooperate.

The first question which must be addressed is how to define cooperation. In economics literature, a cooperative game is one in which groups of players can join together to collectively receive payoffs, which are then apportioned according to the distribution which was agreed to when the group was formed. This leads to some useful definitions of fairness, but it assumes the existance of an infallible contracting agency, which can monitor and enforce agreements between agents.

This paper will treat cooperation as a situation in which two or more agents act in a way which strictly increases their payoffs above what they could achieve if they were to act in a purely self-interested way. Note that this distinguishes *cooperation* where agents actually sacrifice for each other, for example by expending their own resources to do a job for another agent, from *coordination*, where agents interests are naturally aligned, but they need to figure out how to act collectively.

So, when should you cooperate? When should you try to increase the payoff of another agent at the expense of your own payoff? (If you aren't sacrificing your own payoff, you are still acting in a self-interested manner, and therefore are not cooperating). Note that it is necessary to distinguish between a self-interested agent from an agent which wants another agent to do well. Clearly, an agent which wants its opponent to do well should be willing to sacrifice its own payoff in order to increase the payoff of the other agent. But when should a self-interested agent choose to sacrifice its own payoff for its opponent? It should do so when it believes that its opponent will reciprocate the sacrifice to the benefit of both agents. For example, if two self-interested agents each have the opportunity to sacrifice one dollar to give the other agent two dollars, and they each do so, both will profit. Note that it is necessary to interact with another agent multiple times in order to develop a rational expectation that that other agent will reciprocate your cooperative behavior.

The most difficult question is how do you cooperate. In some environments, this question is trivial. For example, in Prisoner's Dilemma cooperative actions are clearly labelled as such. Unfortunately, this information is not available in many environments. The wide variety of potential environments makes it difficult to come up with a method of cooperation which can be applied to every environment.

This paper describes an environment designed to study cooperation. A unique feature of this environment is that it does not provide repeated exposure to a single game but only repeated interactions with the same player. To ensure that strategies developed for this environment are suitable for a wide variety of environments, we have created an environment with as few constraints as possible.

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.

In Section 2 we discuss characteristics of the environment necessary to allow for cooperation between agents, and desirable characteristics of cooperating agents. In Section 3 we discuss related work in cooperation and learning. In Section 4 we describe the environment we use to explore cooperation. In Section 5 we describe a method of achieving cooperation in that environment. In Section 6 we present our conclusions and discuss possible areas of future research.

2 Requirements for environment and players

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 the nature of 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.

We cannot use constant-sum normal form games, since if the sum of the agents payoffs at each outcome is a constant neither player can gain except at the expense of the other. In that situation cooperation is impossible, so the games in our environment must be general sum. Also, the payoffs cannot be identical for all outcomes, since that would make the actions of a self-interested agent indistinguishable from those of a cooperating agent. 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.

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.

3 Related Work

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 [Nowak, 2006] 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, or variants such as win-stay lose-shift [Nowak and Sigmund, 1993].

Research in multi-agent systems has shown that when agents cooperate the social welfare increases. We use the concept of attitude to achieve cooperation. Ways of using attitude to explain the cooperation of people when playing certain types of games are described in [Levine, 1998, Rabin, 1993, Sally, 2002]. 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 *et al.*, 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 *et al.*, 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 and 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 *et al.*, 2003]), but most focus on achieving the best individual payoff and not on cooperation, and they require that the environment consists 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 and 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 *et al.*,

Player 1	Player 2		
	Rock	Paper	Scissors
Rock	0 0	-1 1	1 -1
Paper	1 -1	0 0	-1 1
Scissors	-1 1	1 -1	0 0

Figure 1: Rock-Paper-Scissors as normal form game.

Player 1	Player 2	
	Cooperate	Defect
Cooperate	3 3	0 5
Defect	5 0	**1 1**

Figure 2: The Prisoner's Dilemma. The payoffs for the Nash equilibrium are in bold.

2000], which enables coordination between two agents without communication.

4 Environment

We have chosen to use randomly generated normal form games to investigate cooperation, and to use a system of modified games using player attitudes to cooperate in that environment. This raises two closely related questions. What effect does the nature of the random distribution of the games have on cooperation, and what effect do various values of player attitude have on the payoffs to the players? To investigate them, we have generated games from a wide variety of distributions, and observed the average payoffs for a wide variety of attitude values.

A normal form game is a way to express an interaction between two or more players. A normal form game consists of a set of players, a set of actions for each player, and a payoff function for each player which maps combinations of actions to outcomes for that player. Figure 1 shows the game Rock-Paper-Scissors expressed as a normal form game.

In order to avoid falling into the trap of developing a method of cooperation which only works for a single game, we use randomly generated games. They are more complex than a simple game like Rock-Paper-Scissors, but it is still possible to analyze them in a reasonable amount of time. Figure 3 shows a typical example of the randomly generated games that we use.

The standard solution concept in normal form games is the Nash equilibrium, a probability distribution over the moves of each player with the property that no player can do better by playing a non-Nash strategy than they do under the Nash equilibrium. This means that if one player plays the Nash equilibrium, the other player will have no incentive to adopt a non-Nash strategy. Unfortunately, the Nash equilibrium can be highly inefficient. For example, Figure 2 shows a game called the Prisoner's Dilemma. The bold outcome is the Nash equilibrium, but it is inferior to the outcome achieved by mutual cooperation.

To achieve cooperation we use a concept which has been used to try to explain cooperative behavior in humans. To cooperate in a particular game agents adopt positive attitudes towards each other. An agent's *attitude* is a real number indicating how much the agent values the payoff of the other agent. An attitude of zero indicates an indifferent agent, an attitude of one indicates a highly cooperative agent, and an attitude of negative one indicates a very hostile agent.

Agents choose their move by creating a modified game where each agent's payoff is equal to its payoff in the original game plus its attitude times its opponent's payoff.

$$Payoff'_{agent} = Payoff_{agent} + attitude_{agent} * Payoff_{opponent} \quad (1)$$

P1	P2															
	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
A	.39 .68	.48 .39	.43 .98	.01 .79	.64 .06	.58 .32	.50 .29	.57 1	.27 .69	.07 .36	.97 .75	.26 .59	.35 .09	.56 .05	.33 .63	.83 .63
B	.87 .04	.30 .94	.24 .15	.34 .20	.76 .56	.04 .30	.66 .13	.43 .69	.71 .72	.72 .90	.64 .06	.73 .74	.53 .56	.11 .99	.35 .15	.73 .08
C	.57 .43	.99 .23	.92 .08	.79 .96	.17 .42	.59 .48	.67 .85	.53 .45	.41 .03	.64 .18	.49 .03	.73 .43	.21 .49	.49 .88	.83 .35	.23 .68
D	.61 .51	.40 .33	.24 .78	.59 .24	.14 .64	.57 .94	.70 .86	.35 .06	.01 .57	.75 .06	.27 .56	.68 .27	.57 .82	.60 .57	.93 .50	.23 .69
E	.62 .17	.63 .20	.28 .37	.48 .89	.29 .05	.50 .41	.27 .02	.99 .88	.32 .95	.24 .99	.56 .27	.63 .45	.37 .14	.36 .92	.85 .48	.56 .79
F	.29 .35	.32 .92	.70 .11	.47 .73	.86 .76	.14 .99	.93 .28	.39 .17	.09 .31	.45 .01	.26 .55	.52 .97	.50 .01	.96 .72	.01 .41	.48 .41
G	.51 .69	.68 .28	.48 .67	.30 .35	.34 .27	.75 .24	.77 .47	.18 .81	.85 .09	.35 .96	.09 .64	.31 .70	.93 .62	.83 .86	.67 .72	.69 .54
H	.32 .38	.04 .13	.88 .49	.34 .48	.44 .27	.92 .94	.99 .29	.62 .11	.46 .06	.10 .88	.89 .27	.95 .19	.86 .35	.33 .08	.87 .56	.69 .03
I	.61 .25	.78 .20	.62 .74	.04 .34	.24 .99	.37 .98	.77 .86	.25 .26	.26 .17	.52 .38	.72 .95	.62 .36	.69 .46	.37 .67	.48 .44	.31 .85
J	.64 .84	.58 .58	.10 .87	.55 .60	.55 .46	.93 .02	.46 .97	.84 .63	.68 .45	.31 .98	.23 .57	.66 .32	.40 .19	.62 .48	.51 .01	.63 .11
K	.50 .68	.30 .65	.84 .52	.71 .41	.87 .63	.64 .13	.44 .52	.16 .47	.43 .75	.13 .92	.69 .82	.17 .73	.94 .85	.49 .86	.72 .22	.95 .03
L	.52 .95	.44 .73	.81 .53	.05 .80	.52 .29	.60 .11	.58 .57	.17 .39	.13 .43	.42 .48	.68 .29	.85 .77	.90 .83	.17 .80	.40 .86	.03 .61
M	.98 .61	.28 .29	.62 .32	.44 .19	.57 .35	.01 .95	.67 .88	.07 .24	.97 .57	.19 .20	.50 .35	.50 .55	.38 .05	.34 .82	.29 .52	.96 .83
N	.10 .00	.84 .33	.64 .73	.99 .46	.13 .67	.16 .49	.81 .52	.21 .79	.35 .37	.66 .35	.33 .25	.46 .92	.52 .67	.21 .72	.02 .23	.26 .83
O	.05 .14	.96 .97	.93 .99	.76 .02	.46 .91	.08 .13	.19 .41	.45 .73	.98 .62	.03 .25	.50 .10	.49 .09	.36 .27	.66 .06	.36 .47	.56 .92
P	.09 .66	.86 .58	.87 .02	.41 .09	.21 .74	.37 .57	.58 .34	.61 .37	.99 .22	.23 .97	.38 .77	.23 .60	.53 .77	.80 .78	.41 .13	.41 .85

Figure 3: A randomly generated normal form game with 16 moves per player. The row player's payoffs are shown on the left. Nash equilibria are:
Player 1 (F(28.1%), G(21.5%), O(15.3%), P(35.1%)) Player 2 (b(29.7%), e(11.3%), n(50.5%), p(8.4%))
Player 1 (O(100%) Player 2 (c(100%))
Player 1 (C(4.1%), D(26.9%), H(40.6%), J(28.5%)) Player 2 (c(4.7%), f(51.2%), g(8.3%), j(35.9%))
Player 1 (C(30.1%), F(52.1%), G(4.7%), O(13.1%)) Player 2 (b(9.1%), d(31%), f(13.2%), n(46.7%))
Player 1 (D(8.1%), F(33%), G(14.7%), H(13.3%) J(30.9%)) Player 2 (f(20%), g(11.6%), j(10.6%), l(22.4%), n(35.3%))
Player 1 (C(27.1%), H(36.4%), K(8.9%), O(27.6%)) Player 2 (c(82.1%), e(10.7%), f(4.9%), j(2.3%))

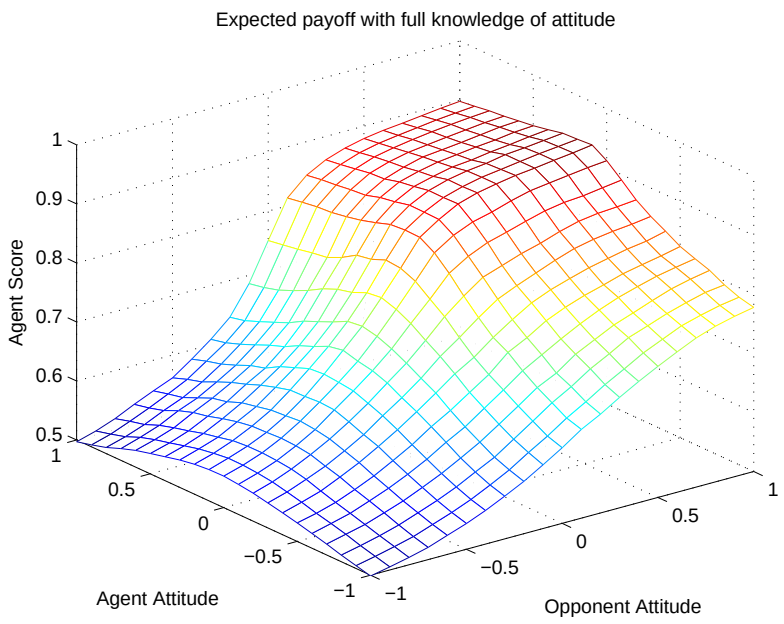

Figure 4: This graph shows the effect of using various attitude values. Results aggregated over 1000 games with payoffs drawn from a uniform distribution between 0 and 1.

Agents find the Nash equilibrium of the modified game, select their move in the original game according to the Nash equilibrium of the modified game, and receive the payoffs from the original game. We have found that by cooperating in this manner, agents can increase their payoffs in the original game. Figure 4 shows the expected payoff in the original game when agents select their action according to this strategy.

A player's payoff is largely dependent on its opponent's attitude, but it can also benefit by increasing its own attitude to a certain degree. If both players adopt a positive attitude, they can collectively improve their payoffs in the game. Note that attitudes can take values above 1 and below negative 1. Figure 5 shows a 16 move game with attitudes varying from -4 to 4. We have chosen to stick to -1 to 1 because values outside that range describe an agent which cares more about its opponent's score than its own score, which seems irrational.

4.1 Number of Moves

The number of moves affects the complexity of the game. Figure 6 shows the results of varying the number of moves from 2 to 16 when the payoffs are drawn from a uniform random distribution from 0 to 1. The z axis shows the agent's score in the original game, averaged over 1000 random games. The x and y axes show the attitudes chosen by the agent and its opponent.

When the number of moves in the game increases, the opportunities for cooperation increase as well. When the game only has 2 moves, the strongest affect on an agent's score is the attitude of its opponent. An agent can do slightly better by adopting an indifferent attitude but the effect of that is dwarfed by the effect of the opponents attitude. As the number of moves increases, we see the scores of the agent increase as well. This is an effect of the greater number of randomly generated payoffs - with more payoffs drawn, there are more chances to draw high payoffs. More interestingly we note that while indifference continues to be a good option, especially with a helpful opponent, the penalty for adopting a negative attitude is much greater than the penalty for adopting a positive attitude. At 16 moves, it is even beneficial for an agent to adopt a slight positive attitude whenever the attitude of its opponent is 0 or more. Most importantly,

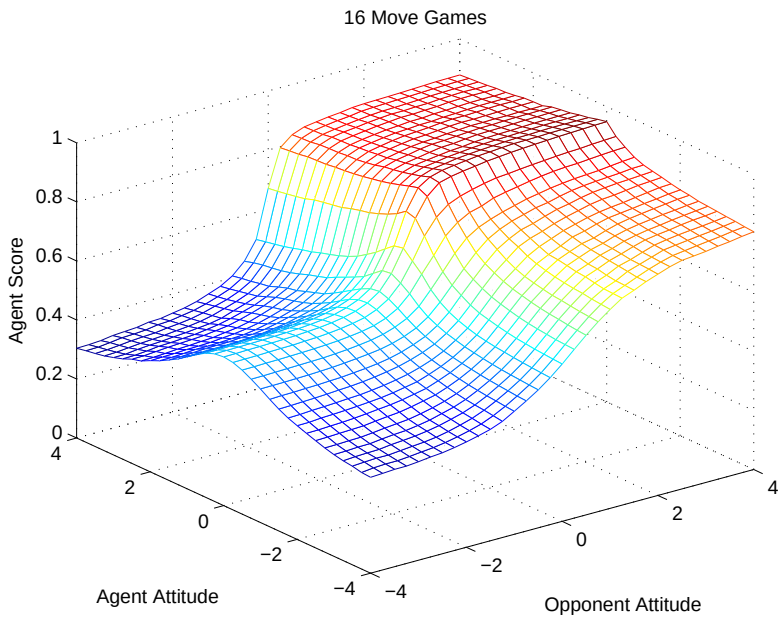

Figure 5: This graph shows the effect on an agents payoff of a wider range of attitudes. Results aggregated over 1000 games with 16 moves per game and payoffs drawn from a uniform distribution between 0 and 1.

this graph shows that an agent does not have a strong incentive against cooperation when its opponent is cooperating.

Figure 7 shows the effect of increasing the number of moves beyond 16. The computing time required to solve the Nash equilibrium increases exponentially with the number of moves. As the number of games increases beyond 16 the plateau of cooperation rises, but the general shape of the graph does not change. We have focused on games with 16 moves, as they provide sufficient complexity to allow for cooperation, while still being computationally tractable.

4.2 Payoff Distribution

Another parameter which can be varied is the random distribution from which payoffs are drawn. Figure 8 shows the effect of using a Gaussian distribution with mean 0 and standard deviation 1. Unlike a uniform distribution, the Gaussian distribution is unbounded, which means that as the number of moves increases, the average achievable payoff increases without bound. There appears to be much less of an opportunity to cooperate in games with payoffs drawn from a Gaussian distribution. At 16 moves, the graph looks similar to the uniform distribution with 4 moves. Figure 9 shows the effect of increasing the number of moves to 128. Because the Gaussian distribution is unbounded, the maximum payoff increases as the size of the game increases. Unlike games generated from a normal distribution, there is no cooperative plateau in which there is little to no benefit to an agent to decrease its attitude. Adopting positive attitudes still allows both agents to mutually increase their payoffs, but the temptation to defect will generally be stronger than in games generated from a uniform distribution. We have chosen to stick with a uniform distribution because cooperation is more straightforward and the resulting games provide a more complex environment, because the payoffs are not dominated by a small number of outlying values.

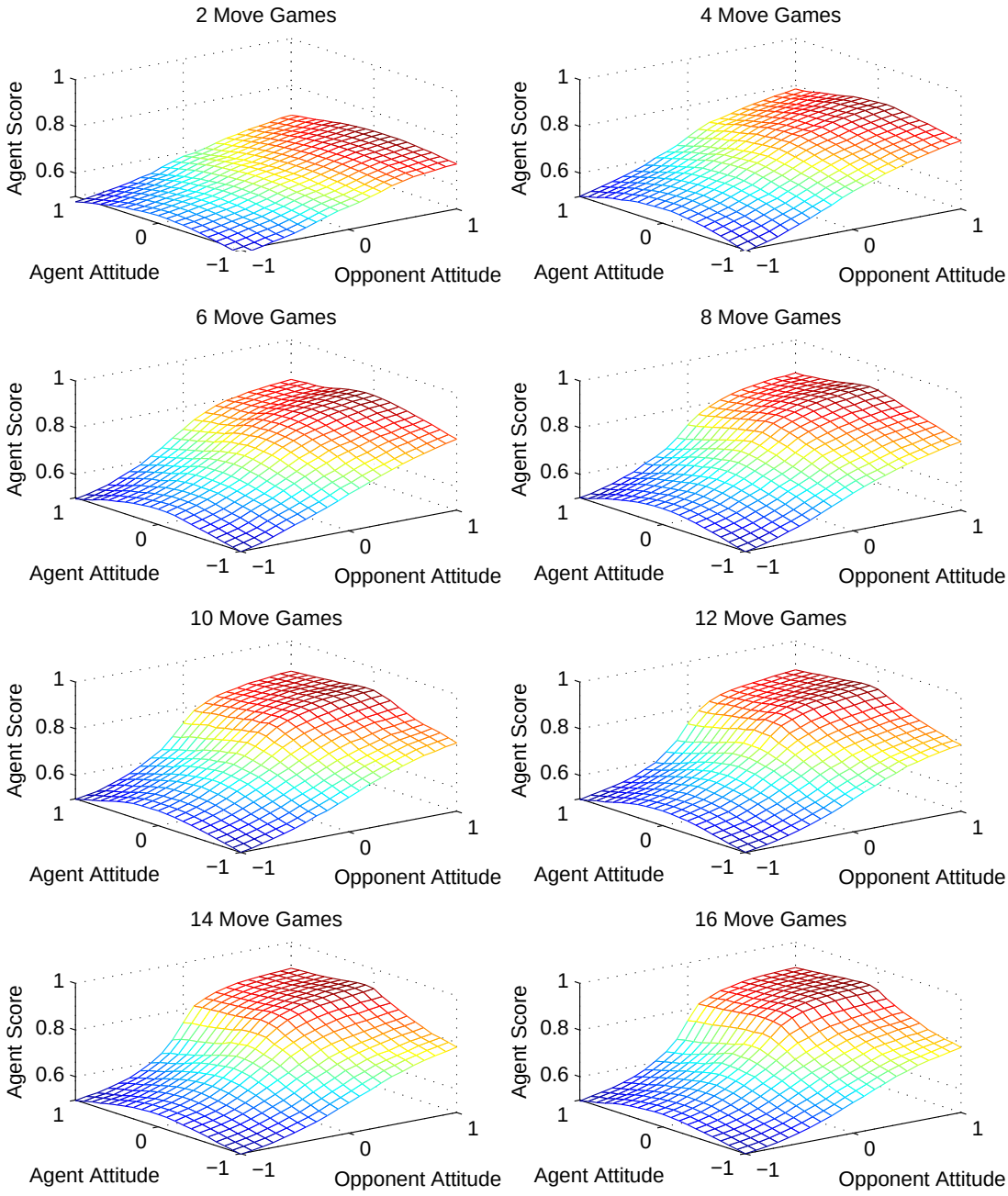

Figure 6: The payoff of an agent is affected by the attitudes of the agent and its opponent. These graphs show how the effects of various combinations of attitudes change as the number of moves in the games increases. These are aggregated results over 1000 games with payoffs drawn from a uniform distribution between 0 and 1.

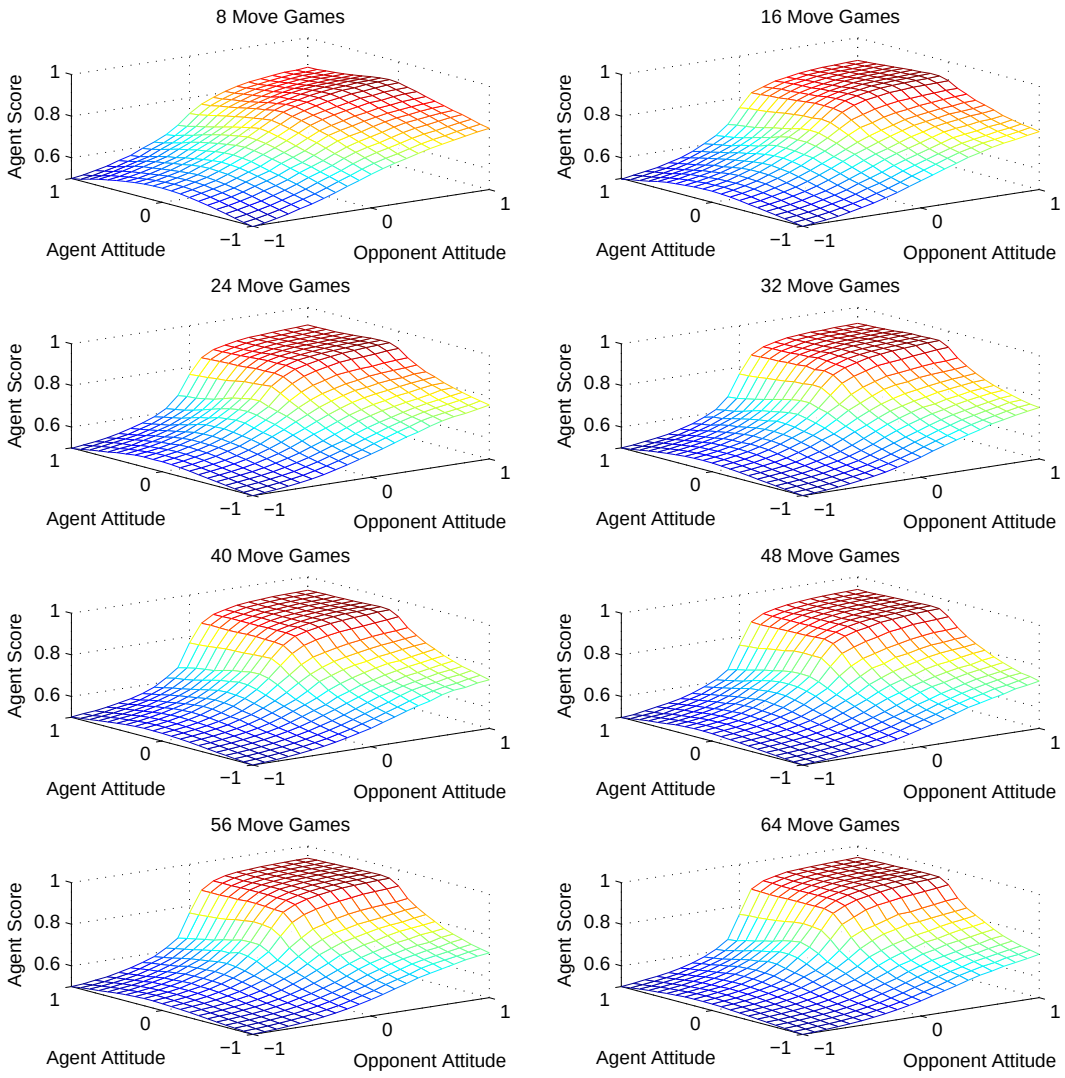

Figure 7: These graphs show the effect of increasing the number of moves beyond 16. Results aggregated over 1000 games with payoffs drawn from a uniform distribution between 0 and 1.

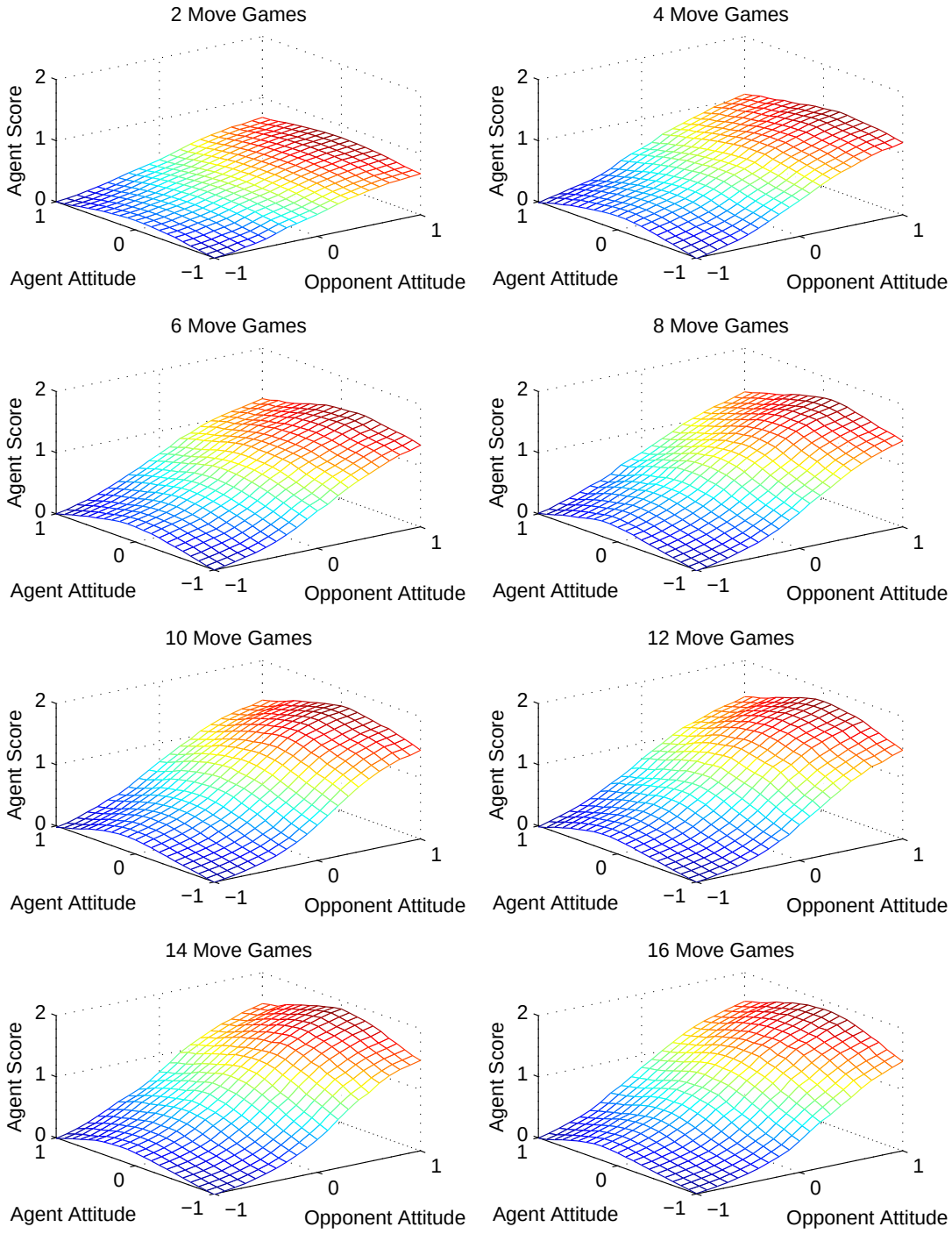

Figure 8: These graphs show the effect of drawing payoffs from a Gaussian distribution. Results aggregated over 1000 games with payoffs drawn from a Gaussian distribution with mean 0 and standard deviation 1.

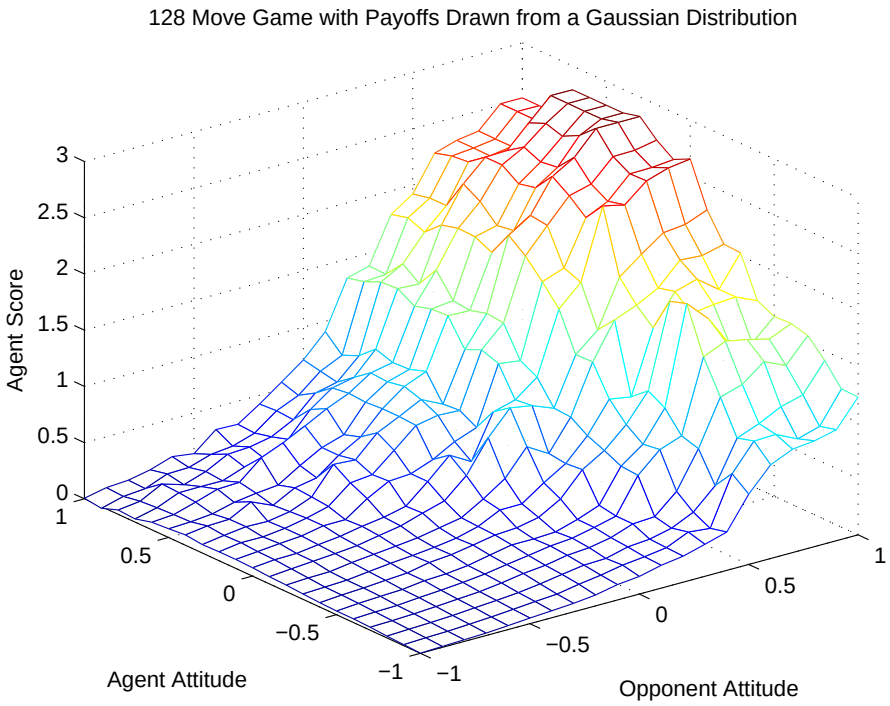

Figure 9: This graph shows the effect of increasing the number of moves to 128. Results aggregated over 12 games with payoffs drawn from a Gaussian distribution with mean 0 and standard deviation 1.

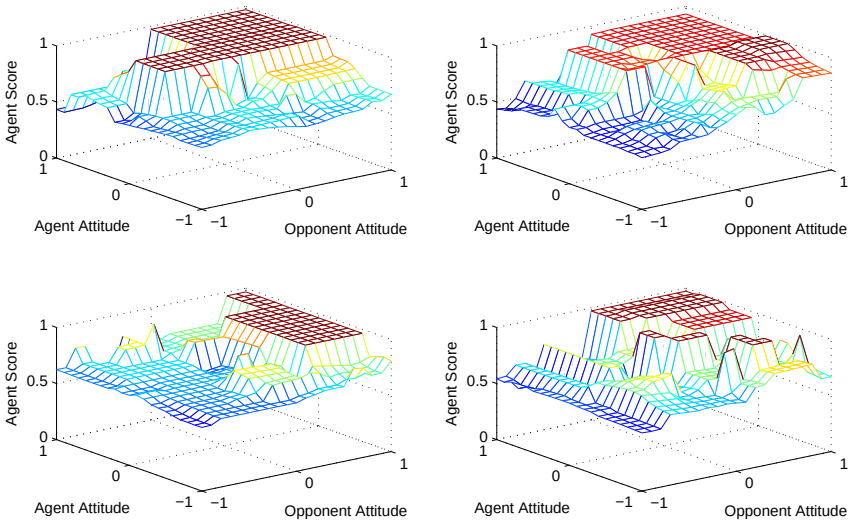

Figure 10: These graphs show 4 examples of the expected payoffs for single 16 move games with payoffs drawn from a uniform distribution from 0 to 1.

4.3 Single Games

The results presented so far are all aggregated over a large number of games. It is important to emphasize that in a specific game, they may be different. Figure 10 shows 4 examples of the expected payoffs for different levels of attitude and belief in a single game. The general shape is similar to the graphs seen earlier, but each individual graph contains pits and outcroppings determined from the specific combinations of payoffs in the game the graph was generated from. Observing different numbers of moves gives some insight into the effect of changing the number of moves. Figure 11 shows a set of graphs generated from games with 2 moves while Figure 12 shows a set of graphs generated from games with 64 moves. Increasing the number of moves increases the complexity of the game, which allows the resulting expected payoff to more accurately reflect the dynamics of cooperation using attitudes.

4.4 Move Inequality

In this environment you can view the number of moves in a game as a measure of its complexity. Another approach is to look at the number of moves available to a player as a measure of the power that player has. Figures 13 and 14 show the effect of an agent having more or fewer moves than its opponent. Unsuprisingly, the impact of the number of moves available to an agent is strongest when its opponent's attitude is negative. In general, having more moves available is beneficial, and having fewer moves available is harmful, but there are 2 exceptions. If the agent's attitude is negative but the opponent's attitude is positive, then the agent's score rises as it loses moves. If the agent and the opponent are cooperating, then reducing the number of moves for the opponent reduces the agents score, because it reduces the number of options available to them collectively.

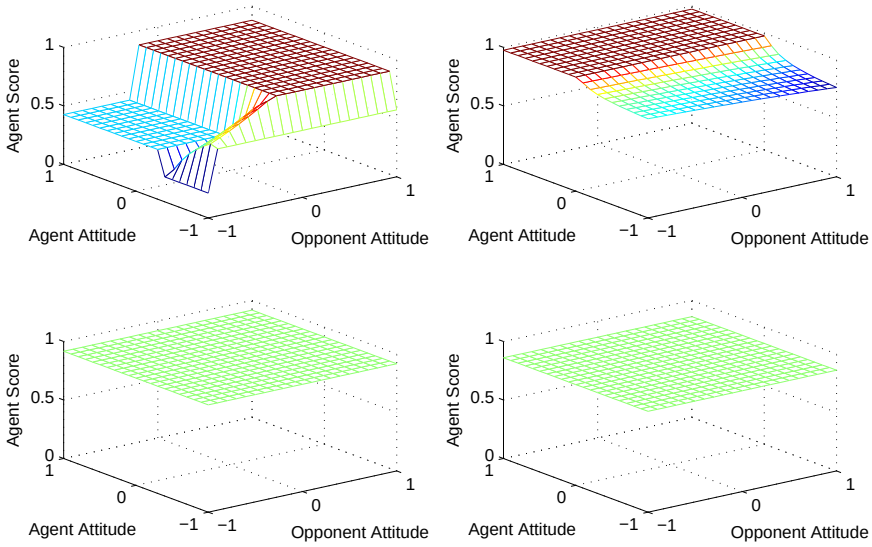

Figure 11: These graphs show 4 examples of the expected payoffs for single 2 move games with payoffs drawn from a uniform distribution from 0 to 1.

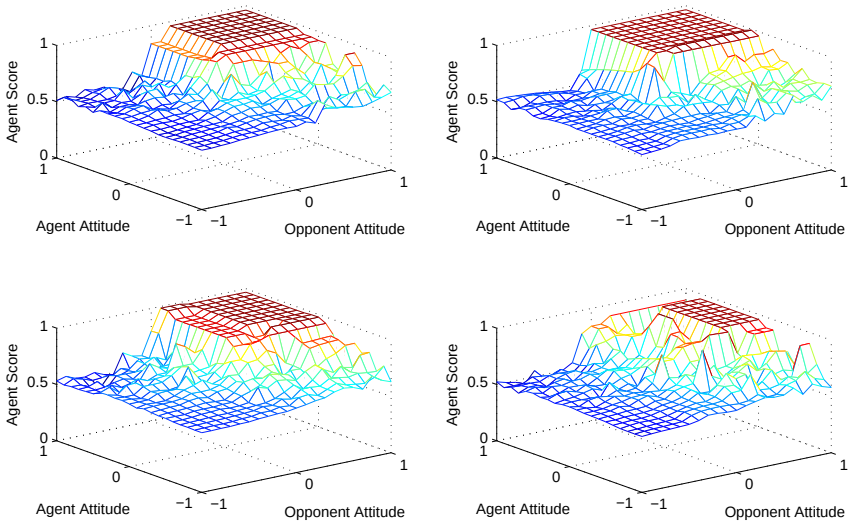

Figure 12: These graphs show 4 examples of the expected payoffs for single 64 move games with payoffs drawn from a uniform distribution from 0 to 1.

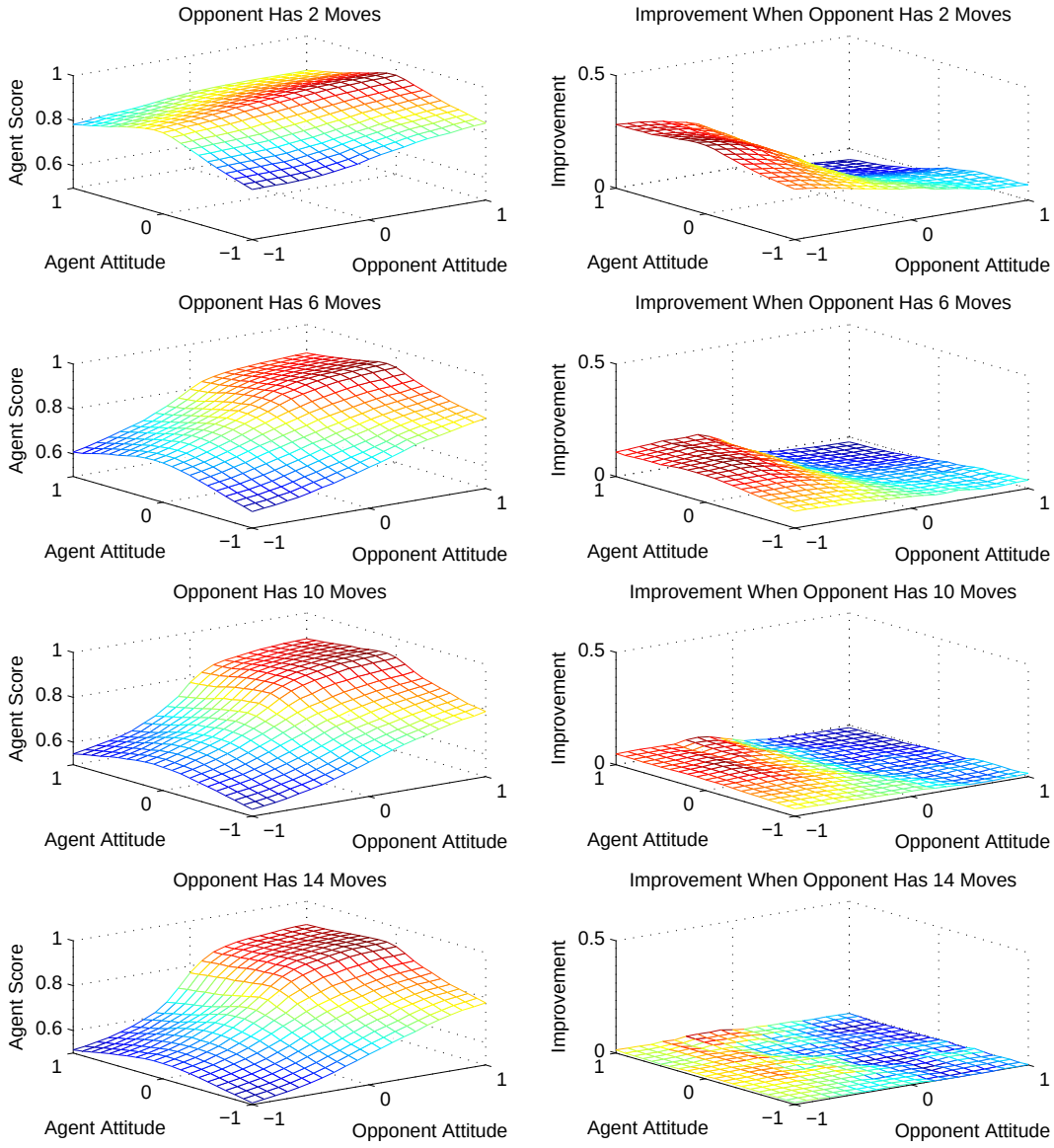

Figure 13: These graphs show what happens when an agent has 16 moves available and its opponent has fewer moves. Graphs on the left show the agent's payoff and graphs on the right show how much the agents payoff improved in comparison to playing against an opponent with 16 moves. These graphs show aggregate results of 1000 games with payoffs drawn from a uniform distribution from 0 to 1.

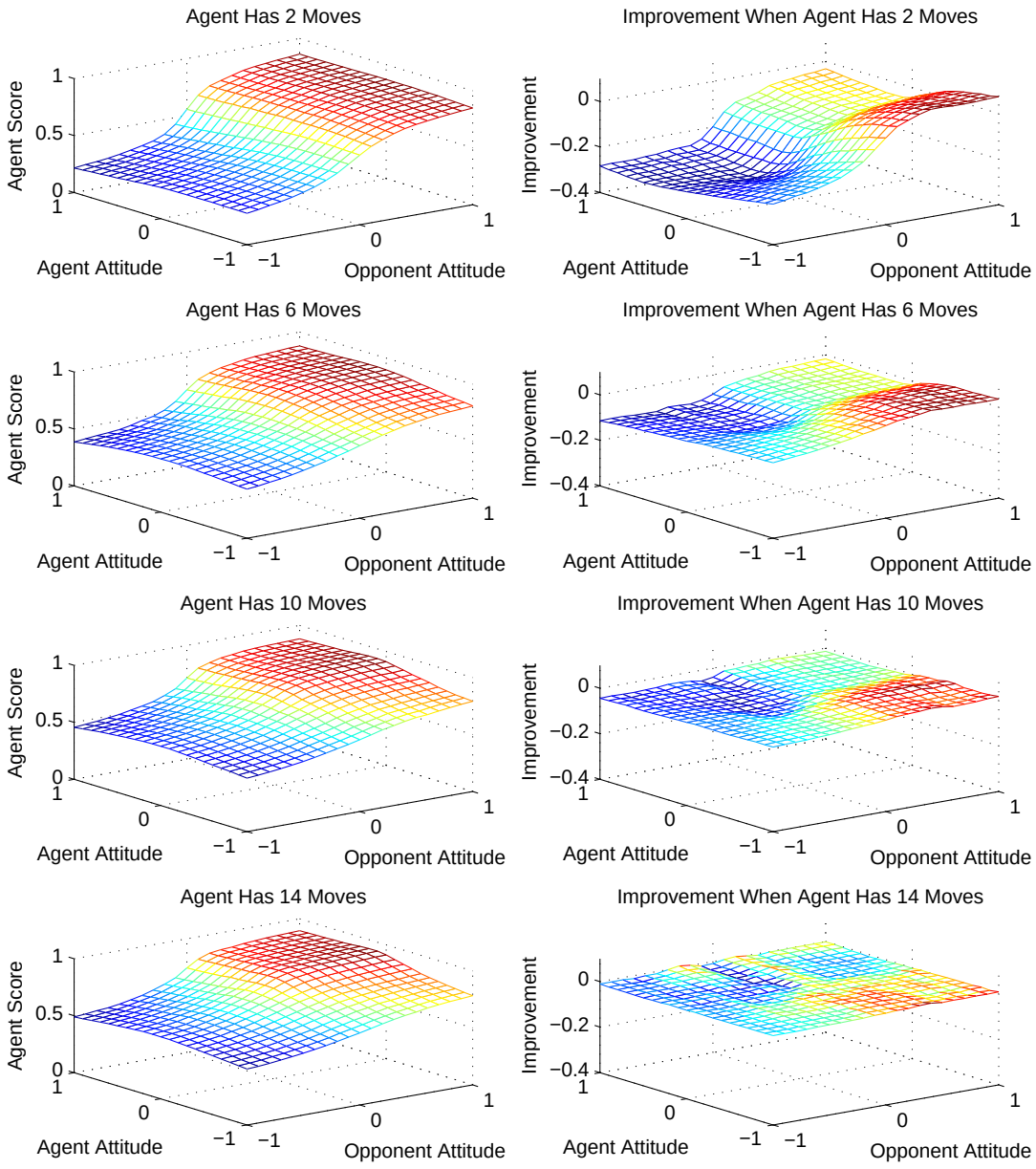

Figure 14: These graphs show what happens when an agent has fewer than 16 moves available and its opponent has 16 moves. Graphs on the left show the agent's payoff and graphs on the right show how much the agents payoff declined in comparison to a game in which it had 16 moves. These graphs show aggregate results of 1000 games with payoffs drawn from a uniform distribution from 0 to 1.

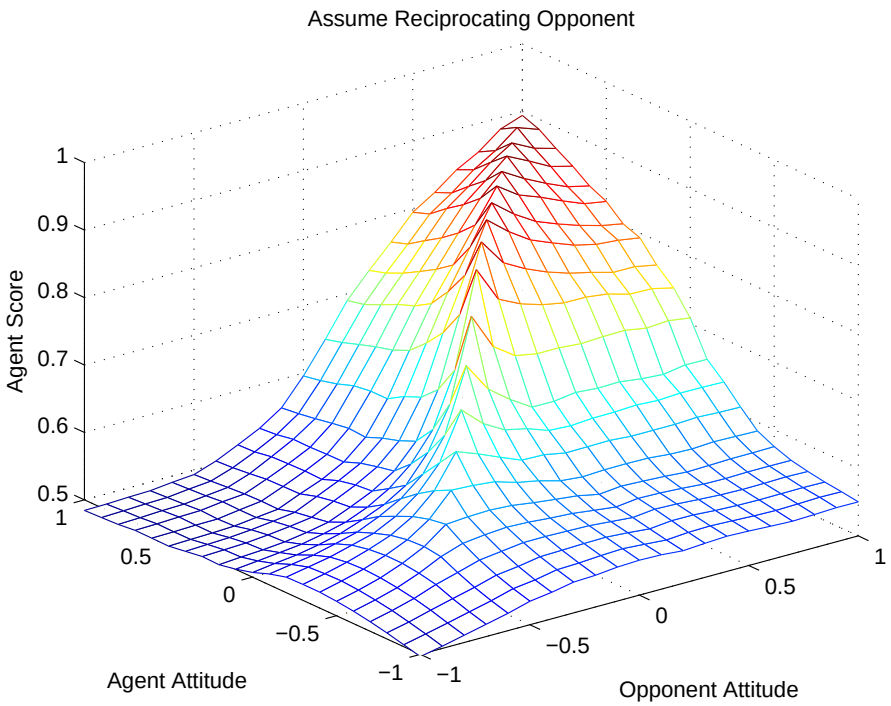

Figure 15: This graph shows an agent's payoff when both players assume that their opponent has the same attitude as them. Results are aggregated over 1000 games, each with 16 moves and payoffs uniformly distributed between 0 and 1.

4.5 Mismatched Attitudes

In previous graphs, the attitude of each player was considered common knowledge. In practical applications agents have no way of knowing what attitude will be used by their opponent. One possible assumption is that the opponent will reciprocate whatever attitude the agent adopts. Figure 15 shows the effect of this assumption on an agents payoff. When each agent assumes that their opponent reciprocates their attitude, they achieve a payoff slightly worse than the payoff they would have received if they both held the attitude of the player with the lower attitude. Another possible assumption is that the opponent is indifferent (has an attitude of 0). Figure 16 shows the effect of that assumption on an agents payoff. When each agent assumes that their opponent is indifferent, then they are best off when their opponent actually is indifferent. Cooperation is impossible because they no longer benefit from the opponent improving its attitude.

It is tempting to have agents get a value for their opponents attitude by having agents disclose their attitudes to their opponent before selecting their moves. Figure 17 shows the problem with this approach. The left graph shows that an indifferent agent can benefit by claiming to have a positive attitude, while the right graph shows that an agent which believes its deceptive opponent will receive a lower payoff. Some lies are not helpful for an agent, such as claiming to have a negative attitude when it is truly indifferent. Some lies do not hurt the opponent, such as claiming to have a negative attitude when the opponents attitude is also negative. However, claiming a positive attitude will always benefit an indifferent agent at the expense of its opponent, so it is not practical to have agents disclose their attitudes to improve coordination.

Both of the assumptions we have examined have proven to be unsatisfactory, and it is not practical to have agents disclose their attitudes because indifferent agents will be motivated to lie. Another possibility to consider is that each agent has an estimate of their opponent's attitude that is generated by adding 0 mean Gaussian noise to their opponent's true attitude. Figure 19 shows the effect of various levels of error

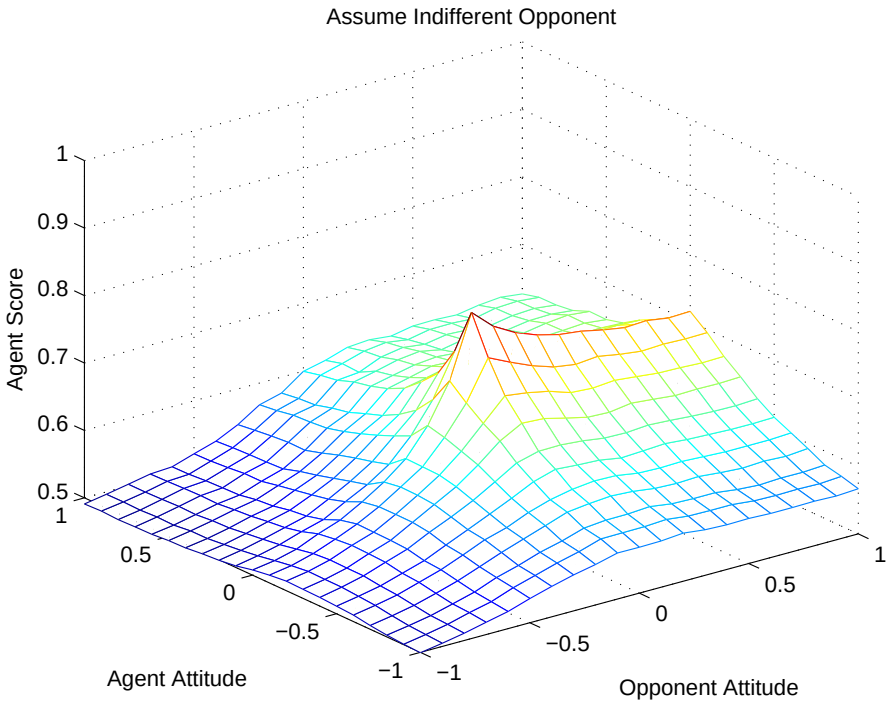

Figure 16: This graph shows an agent's payoff when both players assume that their opponent has an attitude of 0. Results are aggregated over 1000 games, each with 16 moves and payoffs uniformly distributed between 0 and 1.

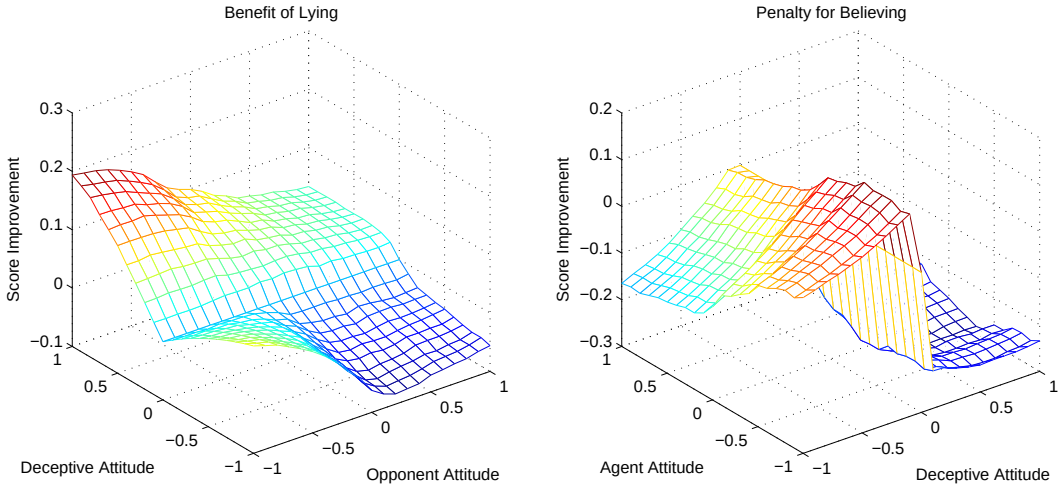

Figure 17: These graphs show the effect of an indifferent agent deceiving a naive opponent. The left graph shows the effect of various false attitudes on the score of the deceptive agent. The right graph shows the effect of various false attitudes on the score of the naive agent. Results are aggregated over 1000 games, each with 16 moves and payoffs uniformly distributed between 0 and 1.

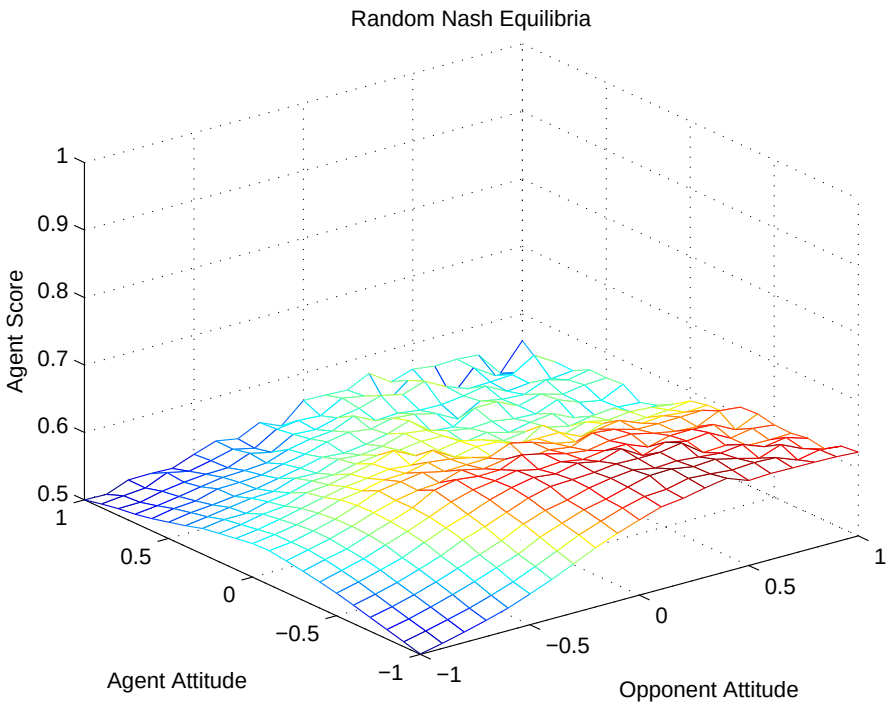

Figure 18: This graph shows an agent's payoff when both players choose a Nash equilibrium randomly instead of selecting the same equilibrium as their opponent. Results are aggregated over 1000 games, each with 16 moves and payoffs uniformly distributed between 0 and 1.

on an agent's payoff. With low levels of error, most of the potential for cooperation remains, but as the level of error increases, the area and the altitude of the plateau are reduced. This means that cooperation is possible with a sufficiently accurate estimate of opponent attitude.

4.6 Mismatched Equilibria

One problem with using the Nash equilibrium to chose a distribution over moves is that there may be multiple Nash equilibria for a game. Randomly generated games with 16 moves per player and payoffs uniformly drawn between 0 and 1 generally have about 8 equilibria. In this situation even when you know the attitude of your opponent, you still need to find which equilibrium it is using. Figure 18 shows the effect of each agent choosing an equilibrium at random. Interestingly, in this situation the lower an agents attitude is, the better it does, especially when its opponent has a positive attitude. This is because agents need to act unpredictably in the presence of hostile agents, so the unpredictability introduced by choosing a Nash equilibrium at random doesn't affect the results as much as it does when the agents are coordinating their actions. Unfortunately, this means that cooperation is impossible when agents choose Nash equilibria randomly. However, there is no reason to select a Nash equilibrium at random. Nash equilibria are found under the assumption that the opponent will be aware of the probability distribution from which the agent selects its move, so it is not rational for an agent using the Nash equilibrium to conceal that probability distribution. We have found a method of learning how an opponent selects a Nash equilibrium which is successful as long as the method used by the opponent is among the methods considered.

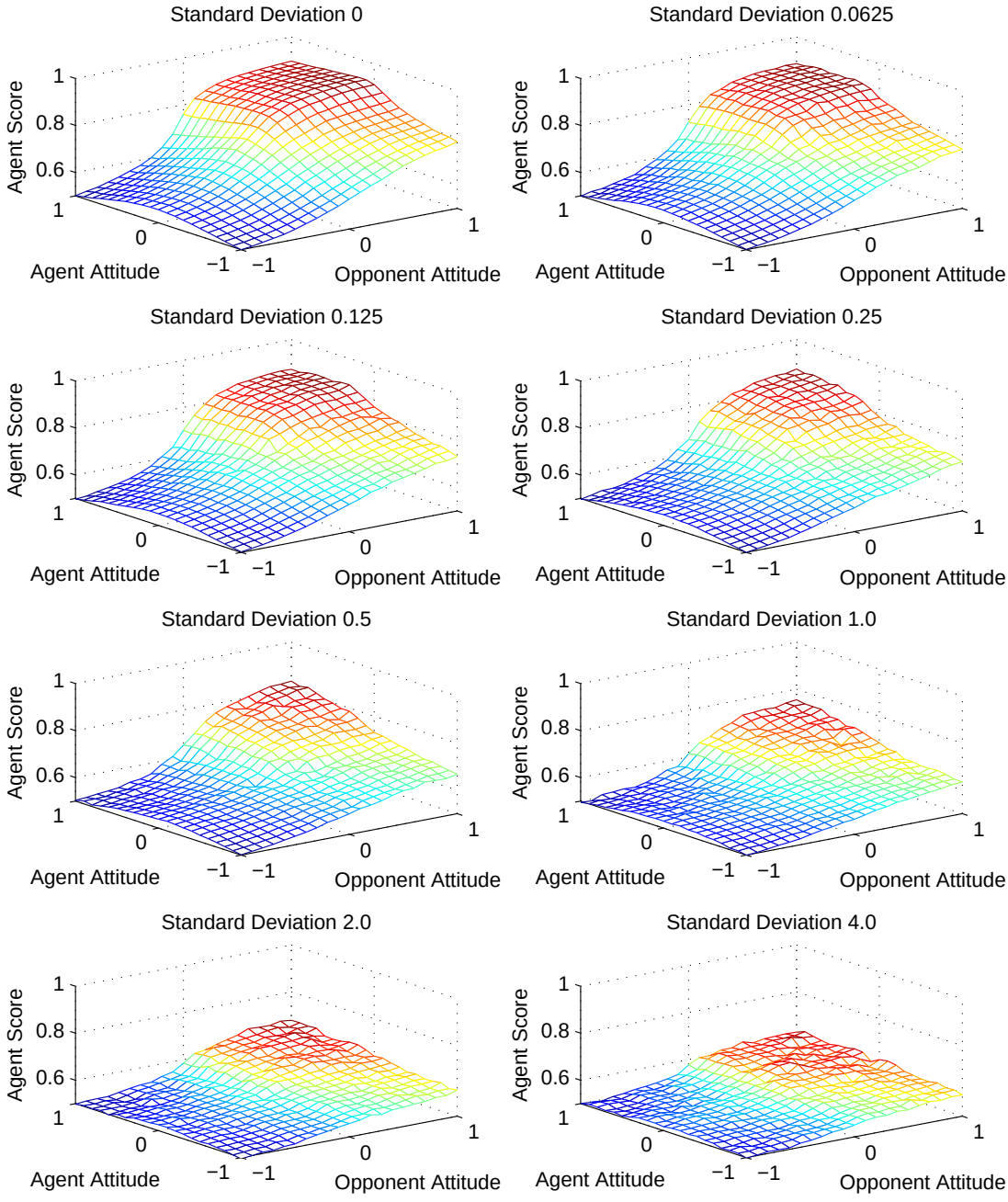

Figure 19: These graphs show the effect of error in agents' estimates of their opponents attitudes. Agents formed an estimate of the opponent's attitude by adding Gaussian noise with a given standard deviation to the opponents true attitude. Agents created a modified game using their own attitude and their estimate of the opponents attitude, and played a Nash equilibrium of the modified game. Results are aggregated over 1000 games, each with 16 moves and payoffs uniformly distributed between 0 and 1.

5 Learning

The method of cooperation we have outlined allows agents to improve their results and is simple enough to be applicable in a wide variety of situations, but it requires knowledge which is not directly available to an agent, namely the attitude of the agents opponent, and how the opponent chooses a Nash equilibrium. In a single game, there is no way for an agent to learn its opponent's attitude, but if an agent has had other interactions with the same opponent, it can use its opponent's actions in the prior interactions as a basis for estimating its opponent's attitude. We will describe a method for estimating the opponent's attitude using Monte Carlo methods to approximate a distribution over the attitude of the opponent.

The evidence available to learn the attitude of the opponent is the opponents actions in previous games. However, the opponents actions are not solely determined by its attitude. In constructing the modified game to chose its action, the opponent will also need to use a value for the agent's attitude. We refer to that value as the *belief* of the opponent. After the modified game has been constructed, the opponent needs to find a Nash equilibrium for the game. Since the game may have multiple equilibria the opponent will need to use some method of choosing specific equilibrium. Unfortunately, there is no standard method of selecting a unique Nash equilibrium. Note that unlike the situation with attitudes, it is not in an agents interests to conceal its method of selecting equilibria. Since an opponent's belief and its method of selecting a Nash equilibrium also affect how it chooses an action, it is necessary to estimate them as well when trying to estimate an opponent's attitude.

Many different methods have been proposed for selecting a unique Nash equilibria [Govindan and Wilson, 2008]. Instead of trying to identify a subset of those as 'reasonable', we have chosen to simulate the problem. The Lemke-Howson algorithm is the method we use for finding Nash equilibria. It can find different equilibria depending on the starting conditions it is given. We use the set of starting conditions for the Lemke-Howson algorithm as a stand-in for the set of 'reasonable' methods of picking a Nash equilibrium. We will show that our algorithm can learn the starting conditions used by the opponent, as well as its attitude and belief.

5.1 Environment

We have chosen to use randomly generated 16-move games with payoffs uniformly distributed between 0 and 1 as the environment within which our agent will attempt to learn the attitude of its opponent. This distribution of games provides a reasonable compromise between the complexity of the game and the computational cost of finding Nash equilibria. Repeated interactions are necessary to estimate the attitude, belief, and starting conditions of an opponent, so we have agents play 1000 different games against each other. We have found that this provides enough time for an agent to converge on an estimate, if it is going to converge.

Here is a brief outline of the environment:

1. Repeat 1000 times:
   (a) Generate a random game with 16 moves per player and payoffs uniformly distributed between 0 and 1.
   (b) Simultaneously show the game to each agent, and have them select a strategy for the game. An agents strategy for a game is a probability distribution over the moves of that game.
   (c) Draw a move for each agent from the probability distribution provided by that agent.
   (d) Reveal the selected moves to both agents, which may then update their estimates of their opponent.

5.2 Algorithm

The method we use to estimate attitude, belief, and starting conditions is similar to a particle filter. We begin by creating a number of particles, each of which has a value for attitude, belief, and a starting condition. Those values are drawn from a prior distribution which consists of a Gaussian with mean 0

and standard deviation 1 for attitude and belief, and a uniform distribution over all possible values for the starting condition. Upon observing a game, we calculate a probability distribution for each particle by creating a modified game using the values for attitude and belief, and calculating a Nash equilibrium for the game using the starting condition. Upon observing the opponent's chosen move we assign each particle a weight equal to the probability it assigned to that move and resample the set of particles according to those weights. Finally we perturb each particle by adding a 0-mean Gaussian to its values for attitude and belief, and possibly changing the starting condition. The standard deviation of the Gaussian and the probability of changing the starting condition are linearly dependent on the current estimate of the level of error.

It is easy to use particles to represent a distribution over attitude, belief, and starting conditions because it is straightforward to calculate $P(move|attitude, belief, startingconditions)$. However, resampling the particles will tend to concentrate all the probability mass to a single point. Perturbation is necessary to avoid that, however it is not clear how to determine the magnitude of the perturbation, especially since there is no model of how an agents attitude and belief may be expected to change. The approach we have chosen to adopt is similar to simulated annealing, in the sense that it gradually reduces the level of perturbation as time goes on. However, unlike simulated annealing, the reduction in perturbation is not strictly determined by the number of steps, but dependent on another value. Intuitively, it makes sense to perturb less when the estimate is accurate and more when the estimate is inaccurate. We have found that this approach works empirically. Naturally an agent isn't aware of it's exact level of accuracy, but it is possible to estimate accuracy sufficiently well.

5.2.1 Estimating Error

In order to determine the level of perturbation, we form an estimate of the error in the current values for attitude, belief, and starting conditions. We define this error as

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

where att_{est} is the average of the attitude over all the particles, and bel_{est} is the average of the belief over all the particles. We can estimate *err* by observing how well the current estimate of attitude and belief predicted the opponents most recent action. An inaccurate estimate is more likely to have predicted the opponents actual move with low probability, or even zero probability.

We estimate error by tracking the relative likelihoods of a set of error values. After observing the opponents move, we calculate the probability assigned to that move, p_{move} by the current estimate of attitude, belief, and starting conditions. Then, given a set of error levels E_l, we update the probability of each error level e_l by:

$$P(e_l|p_{move}) = P(p_{move}|e_l) * P(e_l)/P(p_{move}) \tag{3}$$

$P(p_{move})$ will be the same for all error levels, so we can ignore it and normalize the results. $P(p_{move}|e_l)$ is not calculable, so we approximate it by using a lookup table. The lookup table is created by generating a large number of games, and combinations of attitude and belief, and directly observing the value of p_{move} for various levels of error. The upper left graph in Figure 20 shows the difference that the level of error makes in the frequencies with which various probabilities are observed. The largest effect of the level of error on the calculated probabilities for the observed move is that the probability of observing a move with a calculated probability of 0 rises as the level of error rises.

We improve the accuracy of the error estimate by taking the level of cooperation into account. Agents which are cooperating behave more predictable than agents which aren't, which means they tend to choose equilibria with fewer moves and higher probabilities. The level of cooperation is calculated as the correlation between the scores of the agents in the modified game.

$$\text{coop} = \frac{\text{att} + \text{bel}}{\sqrt{\text{att}^2 + 1}\sqrt{\text{bel}^2 + 1}} \tag{4}$$

Figure 20 shows the effect the level of cooperation has on the distribution of the probabilities of observed moves. If the level of cooperation is high (lower right graph), then the probability of observing a move

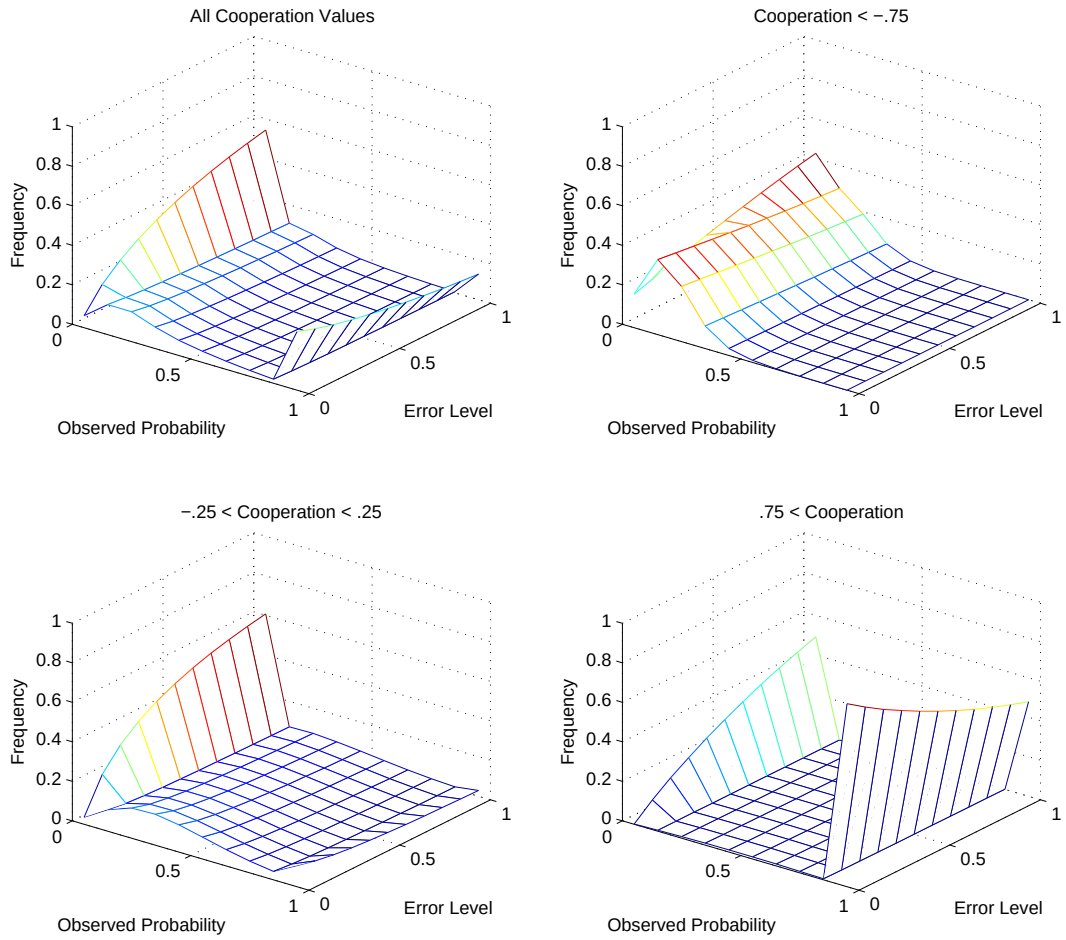

Figure 20: These figures show the distribution of the probabilities assigned to observed moves by an agent with varying levels of error in its estimate of its opponent's attitude and belief. The upper left graph shows the distribution in all conditions. The upper right graph shows the distribution when agents are hostile to each other. The lower left graph shows the distribution when agents are indifferent to each other. The lower right graph shows the distribution when agents are cooperating with each other.

with a calculated probability near 1 is high. If the level of cooperation is low (upper right graph) then the probability of observing a move with a calculated probability near .1 is highest. If the level of cooperation is near 0 (lower left graph) then the calculated probability of the observed move varies more widely, with an average around .4. The accuracy of the error estimate is improved by taking the level of cooperation into account when updating the relative probability of the error levels.

5.2.2 Algorithm Outline

1. Initialize
   (a) Select values for parameters
      i. n = Number of particles (200)

ii. r = Reciprocation level (.1)
iii. f_{ab} = Perturbation factor for attitude and belief (.2)
iv. f_{nash} = Perturbation factor for methods of picking Nash equilibria (.1)
v. $\{e_l | l \in 1..m\}$ = Set of error levels (0,.001,.002,.004,.008,.016,.032,.064,.128,.256,.512,1.0)
vi. $P(e_l)$ = Distribution over error levels (uniform)

(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 4
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.

5.3 Number of Particles

Figure 21 shows the effects of varying the number of particles from 50 to 400. 200 particles is sufficient to learn the opponents attitude and belief, and little improvement is seen as the number of particles increases beyond 200. We use 200 particles for our default setup.

5.4 Perturbation Variations

The question of how to perturb the particles is complicated. In a normal particle filter there is usually a model of how the particles are expected to move, with some fixed level of noise in the model. In this problem domain there is no model of particle movement. An opponents attitude and belief are expected to remain constant. However, simply adopting a fixed level of noise does not result in good performance. We have found that adopting a level of noise with magnitude linearly related to the estimated level of error results in better performance.

5.4.1 Constant Perturbation

The simplest method of perturbing the particles is to add a zero-mean Gaussian with fixed variance to the values for attitude and belief. Unfortunately, this does not result in satisfactory performance. Figure 22 shows the effect of adopting different values for the variance of the perturbation Gaussian. While some values are better than others, none of them result in a reasonable level of performance. If the value is too low, learning happens extremely slowly, and if the value is too high, the system does not converge to a final estimate. The best compromise only achieves a level of error slightly below .2.

5.4.2 Perturbation from Estimated Error

In order to achieve a good estimate in a reasonable amount of time and still converge to a final estimate it is necessary to gradually reduce the magnitude of the perturbation in the particle filter. We have found that one approach which allows the system to converge on a good estimate is to make the magnitude of the perturbation linearly dependent on the error in the current estimate. This allows the system to search a wide area when it has a poor estimate, but as the estimate improves, it focuses its search on the neighborhood of the current estimate. Naturally, the system does not know the exact level of error in the current estimate, but we have found that it is possible to arrive at a reasonable estimate of the current level of error by observing the quality of the predictions in previous games.

Particle values for attitude and belief are perturbed by adding a zero-mean gaussian with standard deviation equal to a constant times the estimated error. Particle values for method of choosing a Nash equilibrium are perturbed by changing the method to a new one with probability equal to the estimated error times a constant.

Figures 23 shows the effect of various combinations of linear constants used to add noise to the system. The best results are achieved when the constant is around .1 to .2. For lower values the level of error drops more slowly, and for higher values the level of error doesn't drop at all.

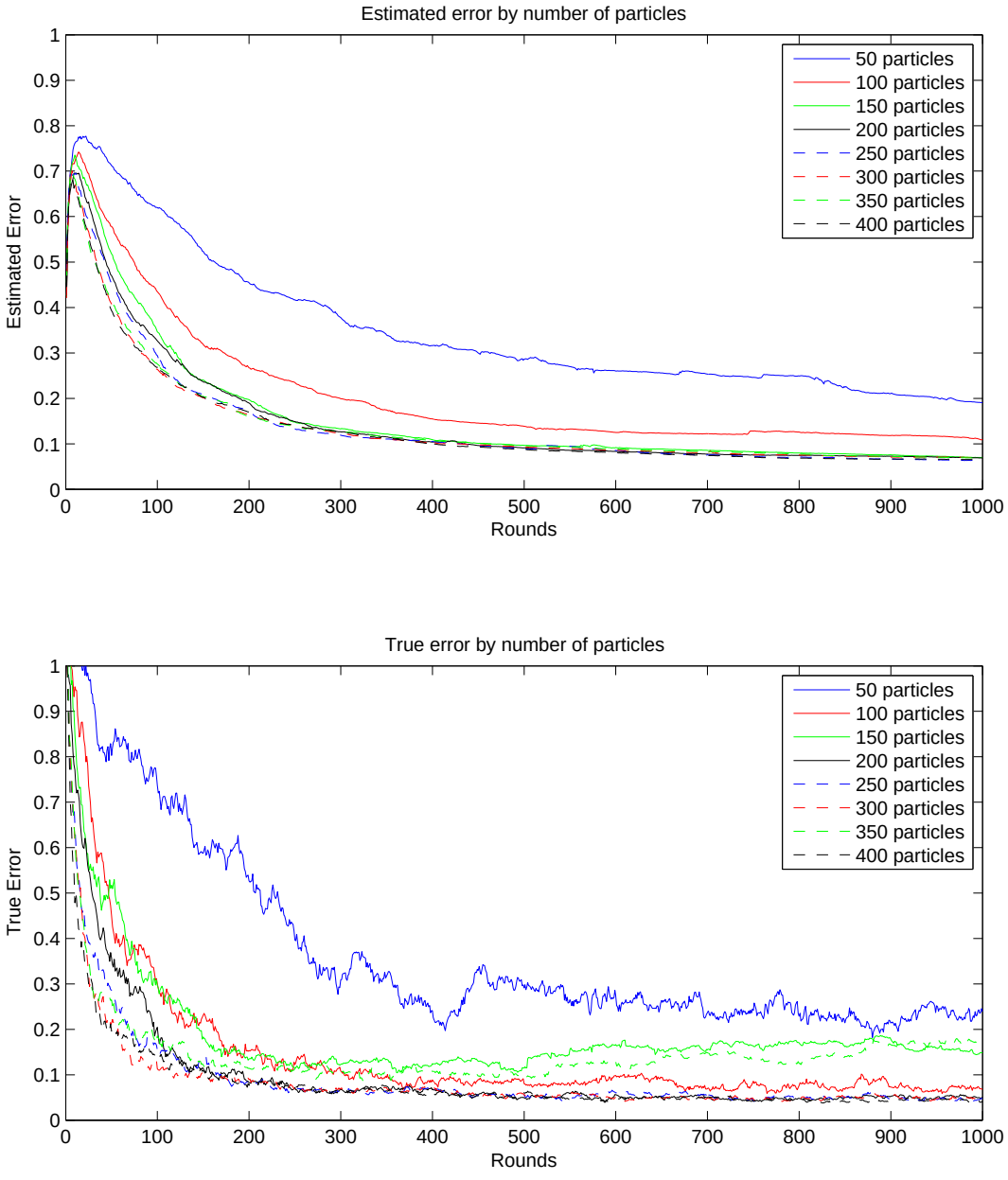

Figure 21: The effect of varying the number of particles on the efficiency of learning. The top graph shows the estimated error, and the bottom graph shows the true error.

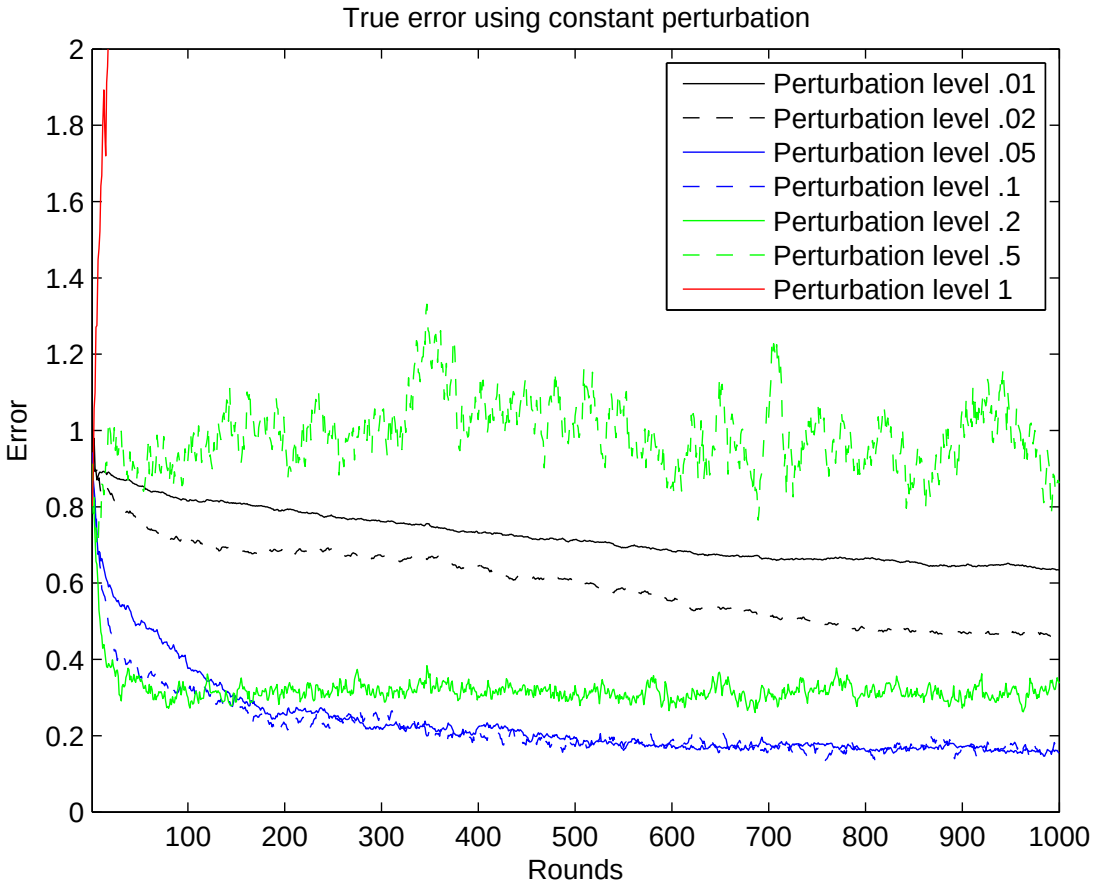

Figure 22: The effect of using a constant level of perturbation.

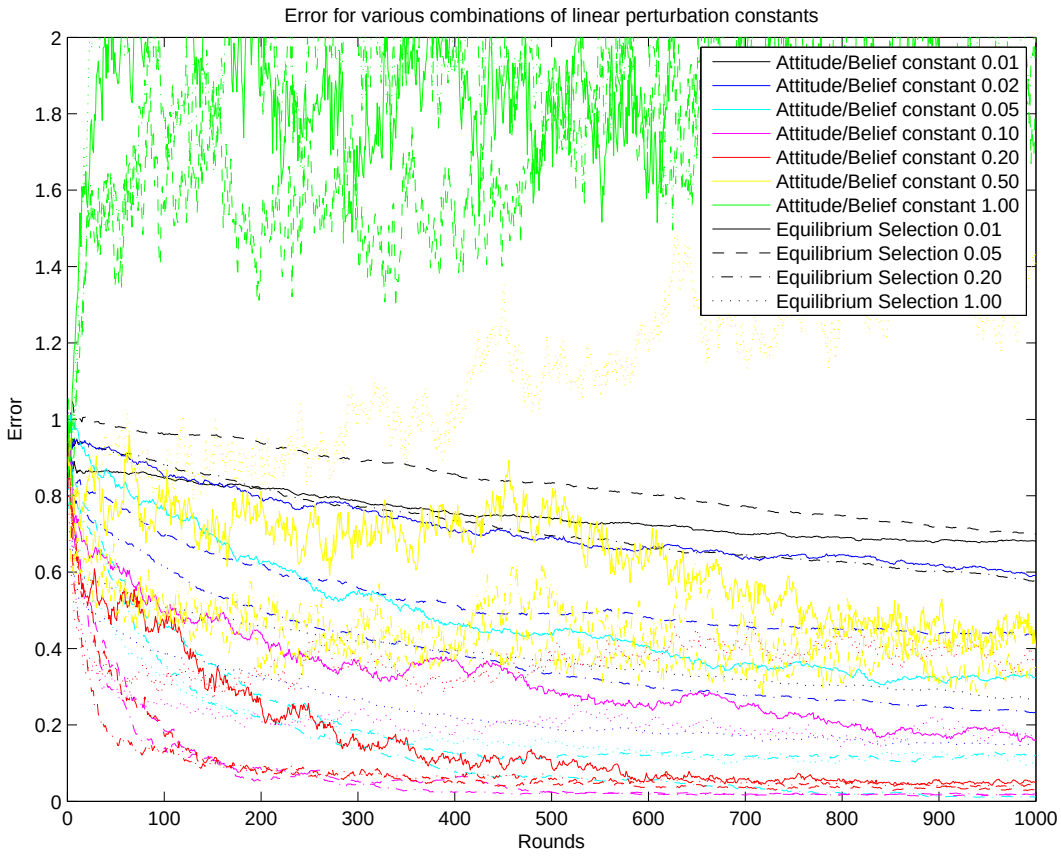

Figure 23: This graph shows the effects of various values for a linear relationship between estimated error and perturbation magnitude.

5.4.3 Discretization Effects

We are using Bayesian reasoning to estimate the relative probability of a set of error levels given the calculated probability for the observed move, and the estimated level of cooperation. Ideally we would have access to a function which provides probability of observing a calculated probability given a level of error and a level of cooperation. Unfortunately we are not aware of such a function, so we have chosen to approximate it by empirically creating a lookup table. To create such a lookup table it is necessary to choose a set of error levels and a partitioning of probability values and cooperation values. The choice of error levels and probability and cooperation partitions can have an effect on the speed and accuracy of the learning algorithm. One obvious area to explore is to vary the number of error levels, probability partitions, and cooperation partitions. Another area to explore is the distribution of those partitions. We have contrasted uniform distributions with distributions with values concentrated close to zero to explore the idea that it may be more important to distinguish between .001 and .002 than between .501 and .502. Figure 24 shows the different sets of values we explored for error levels. We also used those values to partition the set of observed probability values. We divided the set of cooperation values using uniformly sized partitions.

Figure 25 shows the effect of the number and distribution of error values in the lookup table on the speed of learning. In general, increasing the number of error levels tracked improves performance, but not many error levels are required before efficiency gains become negligible.

Name	Values
Geom 5	$0, 3.1e-2, 6.2e-2, 0.125, 0.25, 0.5, 1$
Geom 10	0, 9.7e-4, 1.9e-3, 3.9e-3, 7.8e-3, 0.015, 0.031, 0.062, 0.12, 0.25, 0.5, 1
Geom 15	0, 3.0e-5, 6.1e-5, 1.2e-4, 2.4e-4, 4.8e-4, 9.7e-4, 1.9e-3, 3.9e-3, 7.8e-3, 0.015, 0.031, 0.062, 0.12, 0.25, 0.5, 1
Geom 20	0, 9.5e-7, 1.9e-6, 3.8e-6, 7.6e-6, 1.5e-5, 3.0e-5, 6.1e-5, 1.2e-4, 2.4e-4, 4.8e-4, 9.7e-4, 1.9e-3, 3.9e-3, 7.8e-3, 0.015, 0.031, 0.062, 0.12, 0.25, 0.5, 1
Uniform 5	0, .2, .4, .6, .8, 1
Uniform 10	0, .1, .2, .3, .4, .5, .6, .7, .8, .9, 1
Uniform 15	0, .066, .13, .2, .26, .33, .4, .46, .53, .6, .66, .73, .8, .86, .93, 1
Uniform 20	0, .05, .1, .15, .2, .25, .3, .35, .4, .45, .5, .55, .6, .65, .7, .75, .8, .85, .9, .95, 1

Figure 24: These are values we explored for error levels and to partition probability values.

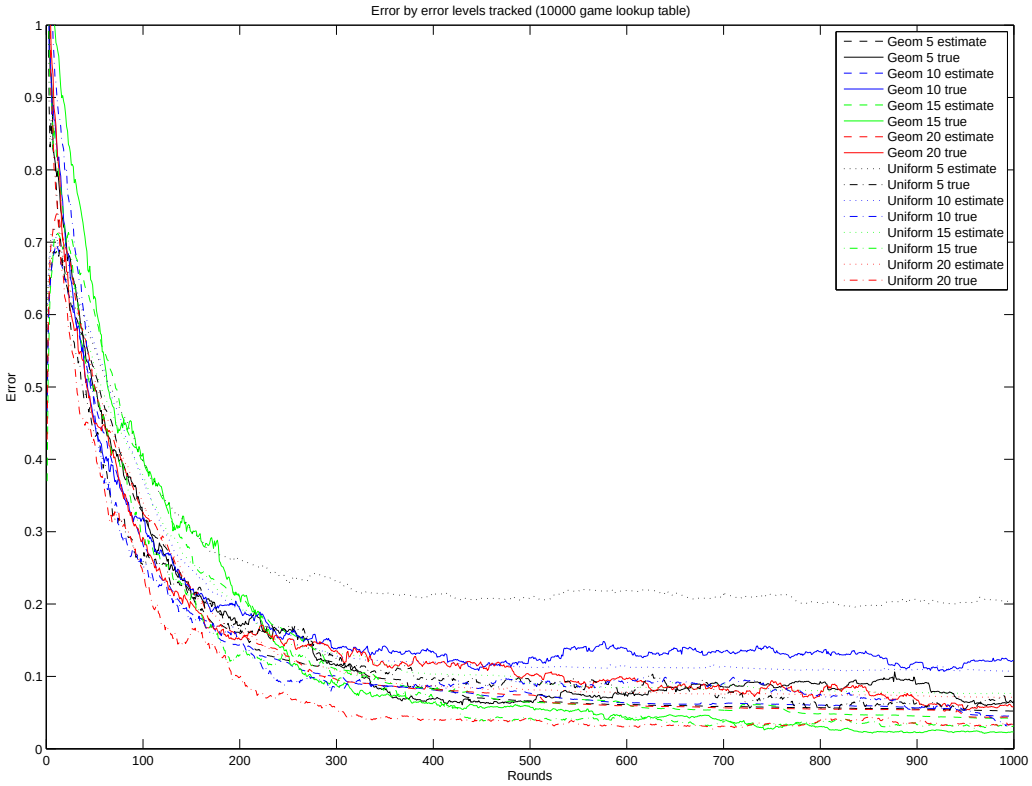

Figure 25: This graph shows the effect of tracking different sets of error levels.

Figure 26 shows the effect of the number of cooperation partitions in the lookup table on the speed of learning. While taking cooperation into account does improve performance, it is not necessary to use a large number of partitions.

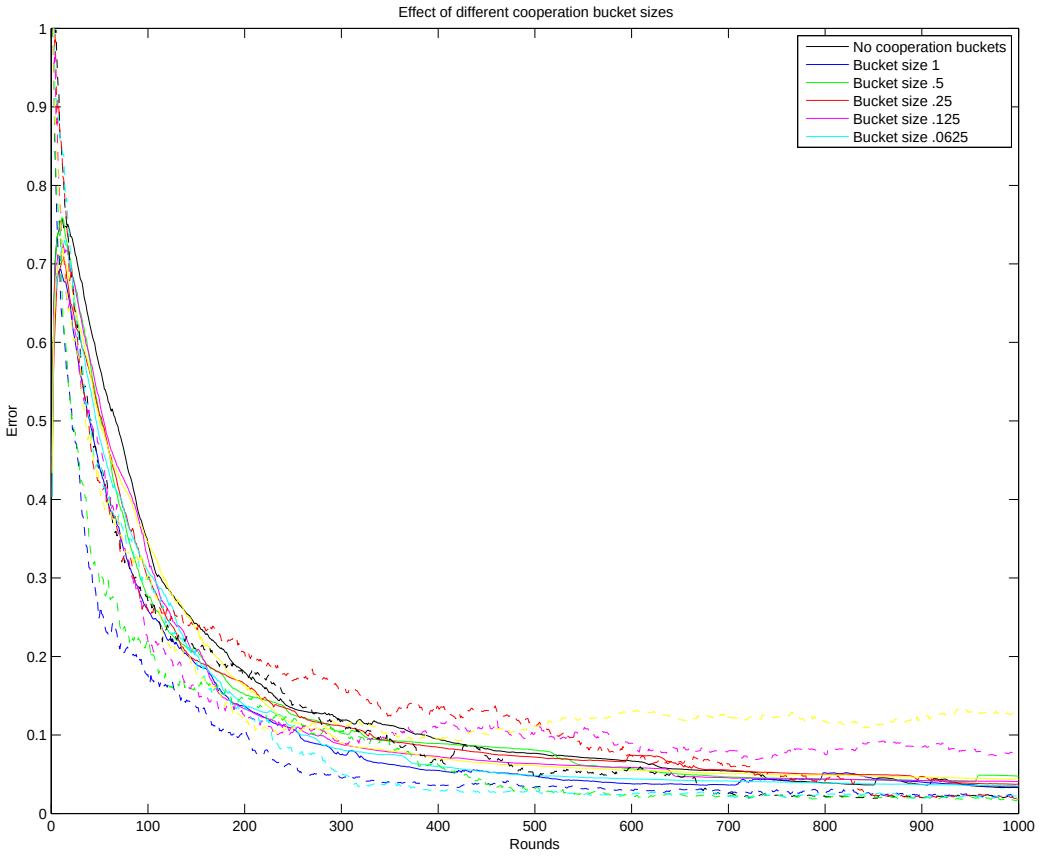

Figure 26: This graph shows the effect of tracking cooperation with different granularities when estimating error.

Figure 27 shows the effect of the number and distribution of probability partitions in the lookup table on the speed of learning. Again, once a reasonable level of tracking is established (about 10 partitions in this case) further increases do not provide a significant advantage.

5.4.4 Limitations to Learning

This system will dependably find the attitude and belief of the opponent, and it will generally find the method used to select Nash equilibria. However, it fails to find the method used to select Nash equilibria if the opponent has an attitude and belief that represent a very low level of cooperation. Figure 28 shows how often the particle filter managed to find the correct method of selecting a Nash equilibria. While it is usually reliable, when the opponent has a negative attitude and belief it usually fails. This is because when both agents have a negative attitude, they act more unpredictably, and it becomes difficult to separate the randomness caused by having the wrong method from the randomness instituted by the opponent to make its actions less predictable.

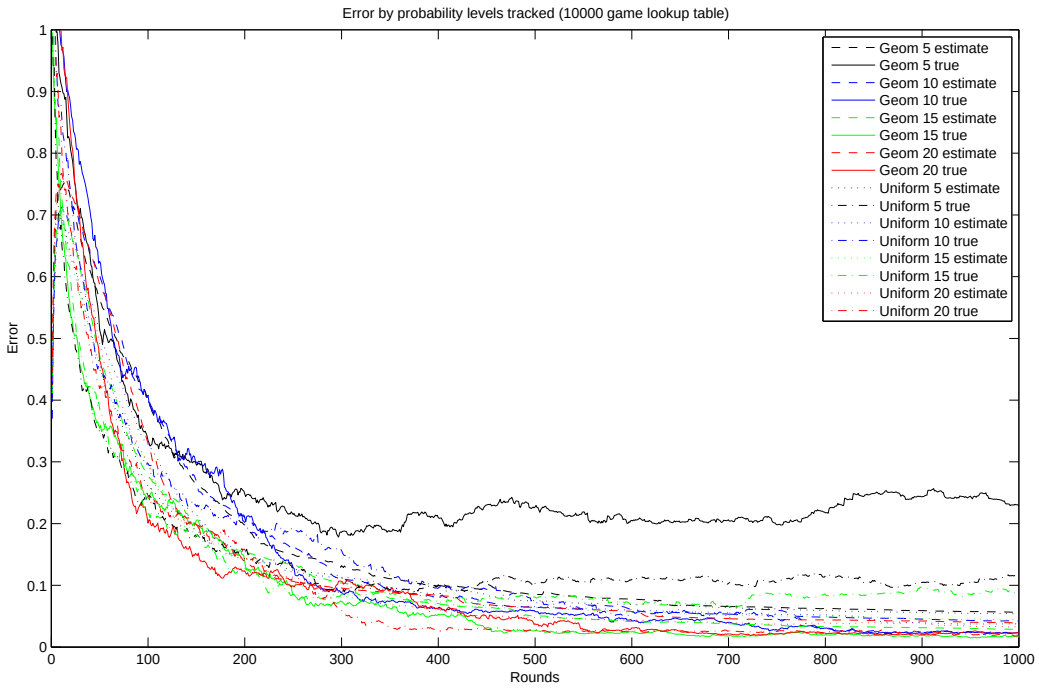

Figure 27: This graph shows the effect of tracking probability with different granularities when estimating error.

6 Conclusions and Future Work

In this paper we have described an environment for studying cooperation and a method of achieving cooperation in that environment which is easily extendable to other environments. We have developed an algorithm which can identify when an opponent is being cooperative or hostile, which allows agents to make informed decisions about whether to cooperate with other agents. While it is not suprising that adopting positive attitudes will help agents improve their score, out results provide a good justification for using attitude.

There are a number of different directions in which this research could be extended. While the learning algorithm provides more than sufficient accuracy to allow for highly effective cooperation, the use of the lookup table in the error estimator requires some prior knowledge of the environment, specifically it requires knowledge of the distribution from which interactions are drawn. A more flexible model in which the agent used the interactions it observed to learn the distribution would be more generally useful.

Our method of achieving cooperation is successful in this environment, and does not expose an agent to undue risk, but it could be improved. For example, if it interacts with a hostile agent, it will never reach the conclusion that cooperation is impossible - it will always try to be nicer than its opponent, even when there are good reasons to believe the opponent will not change. A more sophisticated method of selecting an attitude which takes into account things like the degree of certainty in the current estimate or the degree of certainty the opponent has about its belief may be more successful.

This would also make it easier to apply the learning model to different environments. In randomly generated games, there is usually an opportunity to demonstrate (by selecting a particular move or distribution over moves) that an agent is being slightly cooperative. However, in many games the amount of information revealed by the selected move is not so high. For example, in prisoner's dilemma an agent needs an attitude higher than .66 before it will choose **Cooperation** over **Defect**, which means an agent will not be able

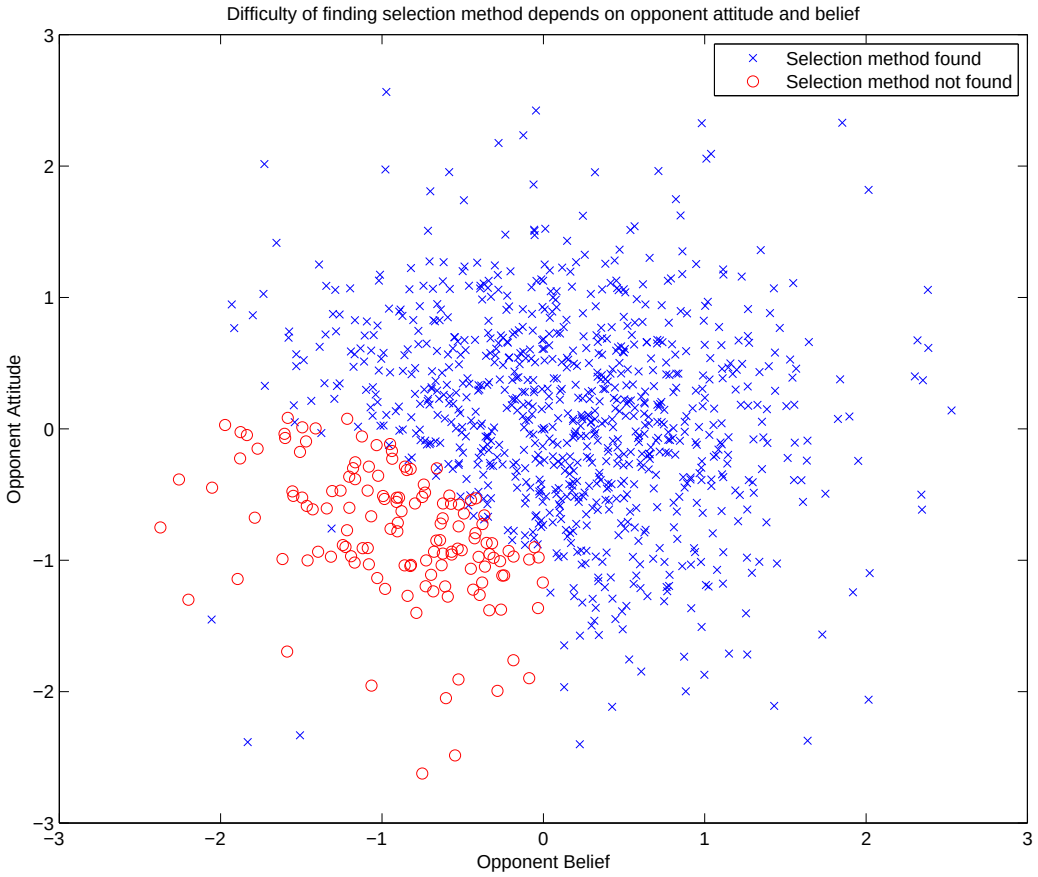

Figure 28: This graph shows how the probability of finding the method used by an opponent to select a Nash equilibrium is affected by the attitude and belief of the opponent.

to tell the difference between a hostile agent and an agent with an attitude as high as .5, which is quite cooperative.

6.1 Acknowledgements

Partial funding from NSF under grant IIS-0414466 is gratefully acknowledged.

References

[Altman *et al.*, 2006] Alon Altman, Avivit Bercovici-Boden, and Moshe Tennenholtz. Learning in one-shot strategic form games. In *Proc. European Conf. on Machine Learning*, pages 6–17. Springer, 2006.

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

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

[Fudenberg and Levine, 1998] Drew Fudenberg and David K. Levine. *The Theory of Learning in Games*. MIT Press, 1998.

[Govindan and Wilson, 2008] Srihari Govindan and Robert Wilson. Refinements of Nash equilibrium. In Steven N. Durlauf and Lawrence E. Blume, editors, *The New Palgrave Dictionary of Economics, 2nd Edition*. Palgrave Macmillan, 2008.

[Kraus *et al.*, 2000] Sarit Kraus, Jeffrey S. Rosenschein, and Maier Fenster. Exploiting focal points among alternative solutions: Two approaches. *Annals of Mathematics and Artificial Intelligence*, 28(1-4):187–258, 2000.

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

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

[Nowak, 2006] Martin A. Nowak. Five rules for the evolution of cooperation. *Science*, 314:1560–1563, 2006.

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

[Saha *et al.*, 2003] Sabyasachi Saha, Sandip Sen, and Partha Sarathi Dutta. Helping based on future expectations. In *Proc. of the Second Int'l Conf. on Autonomous Agents and Multi-Agent Systems*, pages 289–296, 2003.

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

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

[Shoham *et al.*, 2003] Yoav Shoham, Rob Powers, and Trond Grenager. Multi-agent reinforcement learning: a critical survey. Technical report, Stanford University, 2003.

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