

Dispersion Games: General Definitions and Some Specific Learning Results*

Trond Grenager and Rob Powers and Yoav Shoham

Computer Science Department
 Stanford University
 Stanford, CA 94305
 {grenager, powers, shoham}@cs.stanford.edu

Abstract

Dispersion games are the generalization of the *anti-coordination game* to arbitrary numbers of agents and actions. In these games agents prefer outcomes in which the agents are maximally dispersed over the set of possible actions. This class of games models a large number of natural problems, including load balancing in computer science, niche selection in economics, and division of roles within a team in robotics. Our work consists of two main contributions. First, we formally define and characterize some interesting classes of dispersion games. Second, we present several learning strategies that agents can use in these games, including traditional learning rules from game theory and artificial intelligence, as well as some special purpose strategies. We then evaluate analytically and empirically the performance of each of these strategies.

Introduction

A natural and much studied class of games is the set of so-called *coordination games*, one-shot games in which both agents win positive payoffs only when they choose the same action (Schelling 1960).¹ A complementary class that has received relatively little attention is the set of games in which agents win positive payoffs only when they choose distinct actions; these games have sometimes been called the *anti-coordination games*. Most discussion of these games has focused only on the two-agent case (see Figure 1), where the coordination game and the anti-coordination game differ by only the renaming of one player’s actions. However, with arbitrary numbers of agents and actions, the two games diverge; while the generalization of the coordination game is quite straightforward, that of the anti-coordination game is more complex. In this paper we study the latter, which we call *dispersion games* (DGs), since these are games in which agents prefer to be more dispersed over actions.² Although one can transform a dispersion game into a coordi-

	A	B
A	1	0
B	0	1

	A	B
A	0	1
B	1	0

Figure 1: Two-agent coordination game (left) and anti-coordination game (right).

nation game in which agents coordinate on a maximally dispersed assignment of actions to agents, the number of such assignments grows exponentially with the number of agents.

DGs model natural problems in a number of different domains. Perhaps the most natural application is presented by the much studied *load balancing* problem (see, e.g., Azar *et al.* 2000). This problem can be modeled as a DG in which the agents are the users, the possible actions are the resources, and the equilibria of the game are the outcomes in which agents are maximally dispersed. Another natural application of DGs is presented by the *niche selection* problem studied in economics and evolutionary biology. In a general niche selection problem, each of n oligopoly producers wishes to occupy one of k different market niches, and producers wish to occupy niches with fewer competitors. Other niche selection problems include the *Santa Fe bar problem* proposed by Arthur (1994), and the class of *minority games* (Challet & Zhang 1997). These niche selection problems can all be modeled in a straightforward manner by DGs. Finally, we note that DGs can also serve as a model of the process of role formation within teams of robots. In fact, the initial motivation for this research came from empirical work on reinforcement learning in RoboCup (Balch 1998).

This paper makes two types of contributions. First, we formally define and characterize some classes of DGs that possess special and interesting properties. Second, we analyze and experimentally evaluate the performance of different learning strategies in these classes of games, including two standard learning rules from game theory and artificial intelligence, as well as two novel strategies. The remainder of this article is organized as follows. In the first section we present the game definitions. In the second and third sections we present the learning strategies and the results and analysis of their performance. Finally, in the last section we

*This work is supported in part by DARPA Grant F30602-00-2-0598 and by a Benchmark Stanford Graduate Fellowship. Copyright © 2002, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

¹In this paper, we assume familiarity with basic game theory; our formulations are in the style of (Osborne & Rubinstein 1994).

²We chose this name after (Alpern 2001) who studies a subclass of these games which he calls *spatial dispersion problems*.

discuss these findings and present ideas for future research.

Dispersion Game Definitions

In this section we begin by discussing some simple dispersion games, and work our way gradually to the most general definitions. All of the DGs we define in this section are subclasses of the set of *normal form games*, which we define as follows.

Definition 1 (CA, CP, CACP games) A normal form game G is a tuple $\langle N, (A_i)_{i \in N}, (\succeq_i)_{i \in N} \rangle$, where

- N is a finite set of n agents,
- A_i is a finite set of actions available to agent $i \in N$, and
- \succeq_i is the preference relation of agent $i \in N$, defined on the set of outcomes $O = A^n$, that satisfies the von Neumann-Morgenstern axioms.

A game G is a common action (CA) game if there exists a set of actions A such that for all $i \in N$, $A_i = A$; we represent a CA game as $\langle N, A, (\succeq_i)_{i \in N} \rangle$. Similarly, a game is a common preference (CP) game if there exists a relation \succeq such that for all $i \in N$, $\succeq_i = \succeq$; we represent a CP game as $\langle N, (A_i)_{i \in N}, \succeq \rangle$. We denote a game that is both CA and CP as CACP. We represent a CACP game as $\langle N, A, \succeq \rangle$.

Note that we use the notation $\langle a_1, \dots, a_n \rangle$ to denote the outcome in which agent 1 chooses action a_1 , agent 2 chooses action a_2 , and so on. In a CA game where $|A| = k$, there are k^n total outcomes.

Common Preference Dispersion Games

Perhaps the simplest DG is that in which n agents independently and simultaneously choose from among n actions, and the agents prefer only the outcomes in which they all choose distinct actions. (This game was defined independently in (Alpern 2001).) We call these outcomes the *maximal dispersion outcomes* (MDOs).

This simple DG is highly constrained. It assumes that the number of agents n is equal to the number of actions k available to each agent. However, there are many problems in which $k \neq n$ that we may wish to model with DGs. When $k > n$ the game is similar to the $k = n$ game but easier: there is a larger proportion of MDOs. When $k < n$ however, the situation is more complex: there are no outcomes in which all agents choose distinct actions. For this reason, we will need a more general definition of an MDO. In the definitions that follow, we use the notation n_a^o to be the number of agents selecting action a in outcome o .

Definition 2 (MDO) Given a CA game G , an outcome $o = \langle a_1, \dots, a_i, \dots, a_n \rangle$ of G is a maximal dispersion outcome iff for all agents $i \in N$ and for all outcomes $o' = \langle a_1, \dots, a'_i, \dots, a_n \rangle$ such that $a'_i \neq a_i$, it is the case that $n_{a_i}^o \leq n_{a'_i}^{o'}$.

In other words, an MDO is an outcome in which no agent can move to an action with fewer other agents. Note that when the number of agents is less than or equal to the number of actions, an MDO allocates exactly one agent to each action, as above.

Under this definition, the number of MDOs in a general CA game with k actions is given by

$$MDO(n, k) = n! \frac{\binom{k}{n \bmod k}}{[n/k]^{n \bmod k} [n/k]!^k}.$$

When $k = n$ this expression simplifies to $n!$, since there are $n!$ ways to allocate n agents to n actions.

The simple DG presented above also makes another strong assumption. It assumes that an agent's preference over outcomes depends only on the overall configuration of agents and actions in the outcome (such as the number of agents that choose distinct actions), but not on the particular identities of the agents or actions (such as the identities of the actions that are chosen). We call these the assumptions of *agent symmetry* and *action symmetry*. However, many situations we might like to model are not agent and action symmetric. For example, role formation on soccer teams is not action symmetric. The identity of a particular field position in an outcome can affect the performance of the team: a team with a goalie but no halfback would probably perform better than one with a halfback but no goalie, all else being equal. Robot soccer is also not necessarily agent symmetric. If agent 1 is a better offensive than defensive player, then a team may perform better if agent 1 is a forward instead of a fullback, all else being equal. We use the following formal definitions of symmetry.

Definition 3 (Agent Symmetry) A CA game $G = \langle N, A, (\succeq_i)_{i \in N} \rangle$ is agent symmetric iff for all outcomes $o = \langle a_1, \dots, a_i, \dots, a_n \rangle$, and for all permutations $o' = \langle a'_1, \dots, a'_i, \dots, a'_n \rangle$ of o , for all $i \in N$, $o \succeq_i o'$ and $o' \succeq_i o$.

Definition 4 (Action Symmetry) A CA game $G = \langle N, A, (\succeq_i)_{i \in N} \rangle$ is action symmetric iff for all outcomes $o = \langle a_1, \dots, a_i, \dots, a_n \rangle$ and $o' = \langle a'_1, \dots, a'_i, \dots, a'_n \rangle$, if there exists a one-to-one mapping $f : A \rightarrow A$ such that for all $i \in N$, $f(a_i) = a'_i$, then for all $i \in N$, $o \succeq_i o'$ and $o' \succeq_i o$.

In fully symmetric games, agents cannot distinguish between outcomes with the same configuration of numbers of agents choosing actions. Thus we use the abbreviated notation $\{n_1, \dots, n_k\}$ to refer to the set of outcomes in which n_1 agents choose some action, n_2 agents choose a different action, and so on. By convention, we order the actions from most to least populated.

We are now ready to state the formal definition of a weak DG that is well defined over the set of all CACP games, including asymmetric games and games with arbitrary n, k .

Definition 5 (Weak DG) A CACP game $G = \langle N, A, \succeq \rangle$ is a weak dispersion game iff the set of \succeq -maximal outcomes of G is a subset of the set of MDOs of G .

This definition requires only that at least one of the MDOs is a preferred outcome, and that none of the non-MDOs is a preferred outcome. This definition is weak because it places no constraints on the preference ordering for the non-maximally-preferred outcomes.³ For this reason, we also

³The reader may wonder why our definitions don't require that

state a strong definition. Before we can state the definition, however, we will need the following *dispersion relation*.

Definition 6 (\sqsupseteq) Given two outcomes $o = \langle a_1, \dots, a_i, \dots, a_n \rangle$ and $o' = \langle a'_1, \dots, a'_i, \dots, a'_n \rangle$, we have that $o \mathbf{D} o'$ iff there exists a agent $i \in N$ such that $a'_i \neq a_i$, and $n_{a_i}^o < n_{a'_i}^{o'}$, and for all other agents $j \in N, j \neq i, a_j = a'_j$. We let the dispersion relation \sqsupseteq be the reflexive and transitive closure of \mathbf{D} .

In other words, o is more dispersed than o' if it is possible to transform o' into o by a sequence of steps, each of which is a change of action by exactly one agent to an action with fewer other agents. It is important to note that the dispersion ordering is a structural property of any CACP game. The dispersion relation over the set of outcomes forms a partially ordered set (poset). Note that the set of MDOs is just the set of \sqsupseteq -maximal elements of O .

There are many other measures that we could use instead of the qualitative dispersion relation. Entropy is consistent with, but stronger than our dispersion relation: if $o \sqsupseteq o'$ then the entropy of o is higher than that of o' , but the converse is not necessarily true. We have chosen to base our definitions on the weaker dispersion relation because it is the most general, and because it corresponds directly to a single agent's change of actions.

Using this dispersion relation, we can state the formal definition of strong DGs.

Definition 7 (Strong DG) A CACP game $G = \langle N, A, \succeq \rangle$ is a strong dispersion game iff for all outcomes $o, o' \in O$, it is the case that if $o \sqsupseteq o'$ but not $o' \sqsupseteq o$, then $o \succeq o'$ but not $o' \succeq o$.

Recall that the preference relation \succeq forms a total ordering while the dispersion relation \sqsupseteq forms a partial ordering. Thus this definition requires that o is strictly preferred to o' when o is strictly more dispersed than o' .

If the strong definition has such nice properties, why bother to state the weak definition at all? There are many situations which have a dispersion quality but which cannot be modeled by games in the stronger class. Consider the situation faced by Alice, Bob, and Charlie who are each choosing among three possible roles in the founding of a company: CEO, COO, and CFO. Because they will be compensated as a group, the situation can be modeled as a CP game. However, suppose that Bob would be a terrible CEO. Clearly, the agents would most prefer an outcome in which each role is filled and Bob is not CEO; thus the game satisfies the weak definition. However, rather than have all roles filled and Bob alone be CEO, they would prefer an outcome in which Bob shares the CEO position with one of the other agents (i.e., both Bob and another agent select the "CEO" action), even though it leaves one of the other roles empty. In other words, the preference relation conflicts with the dispersion ordering, and the game does not satisfy the strong definition.

all MDOs are maximal outcomes. In fact, it is easy to verify that this must be the case in a fully symmetric DG.

Non-Common-Preference Dispersion Games

There are also several interesting classes of non-CP dispersion games we might like to model. Due to space considerations we will not define these classes formally, but instead present a few motivating examples.

Consider again the load balancing application in which each of n users simultaneously wishes to use one of k different resources. If the users all belong to a single organization, the interest of the organization can be well modeled by a CP DG, since the productivity of the organization will be highest if the users are as dispersed as possible among the servers. However, the users' preferences may be more *selfish*: a user may prefer individually to use a resource with the fewest possible other users, regardless of the welfare of the rest of the group. Additionally, users' preferences may reflect some combination of individual and group welfare. These problems may be modeled with the class of *selfish dispersion games*.

Consider again the niche selection problem, in which each of n oligopoly producers wishes to occupy one of k different market niches. It may be the case that in addition to a general preference for dispersal (presumably to avoid competition) each producer has an *exogenous* preference for one of the niches; these preferences may or may not be aligned. For example, it may be that one of the market niches is larger and thus preferred by all producers. Alternatively, a producer may have competencies that suit it well for a particular niche. Note that the two agent case can be modeled by what one might call the *anti-battle-of-the-sexes* game in which a man and his ex-wife both wish to attend one of two parties, one of which is more desirable, but both prefer not to encounter each other (the reader familiar with the original BoS game will appreciate the humor). These problems can be modeled with the class of *partial dispersion games*, in which agents' preferences may align with either the dispersion ordering or with a set of exogenous preferences.

Learning Strategy Definitions

Now that we have defined a few interesting classes of dispersion games, let us consider the task of playing them in a repeated game setting. There are two perspectives we may adopt: that of the individual agent wishing to maximize his individual welfare, and that of a system designer wishing to implement a distributed algorithm for maximizing the group welfare. In the present research, we adopt the latter.

Let us begin with the problem of finding an MDO as quickly as possible in a weak CACP DG.⁴ Note that this problem is trivial if implemented as a centralized algorithm. The problem is also trivial if implemented as a distributed algorithm in which agents are allowed unlimited communication. Thus we seek distributed algorithms that require no explicit communication between agents. Each algorithm takes the form of a set of identical learning rules for each

⁴Note that any mixed strategy equilibrium outcome is necessarily preference dominated by the pure strategy MDOs. For this reason, we henceforth disregard mixed strategy equilibria, and focus on the problem of finding one of the MDOs.

agent, each of which is a function mapping observed histories to distributions over actions.

Consider the most naive distributed algorithm. In each round, each agent selects an action randomly from the uniform distribution, stopping only when the outcome is an MDO. Note that this naive learning rule imposes very minimal information requirements on the agents: each agent must be informed only whether the outcome is an MDO. Unfortunately, the expected number of rounds until convergence to an MDO is

$$\frac{k^n}{MDO(n, k)}.$$

It is easy to see that for $k = n$ the expected time is $n^n/n!$, which is exponential in n .

We began by evaluating traditional learning rules from game theory and artificial intelligence. Game theory offers a plethora of options; we looked for simplicity and intuitive appropriateness. We considered both fictitious play (Brown 1951; Robinson 1951) and rational learning (Kalai & Lehrer 1993). Rational learning did not seem promising because of its dependence on the strategy space and initial beliefs of the agents. Thus we focused our attention on fictitious play.

In evaluating learning rules from artificial intelligence the decision was more straightforward. Recently there has been significant interest in the application of reinforcement learning to the problem of multi-agent system learning (Littman 1994; Claus & Boutilier 1998; Brafman & Tenenholz 2000). We chose to implement and test the most common reinforcement learning algorithm: *Q-learning*.

Finally, we developed a few special purpose strategies to take advantage of the special structure of DGs.

Note that the different strategies we describe require agents to have access to different amounts of information about the outcome of each round as they play the game. At one extreme, agents might need only a Boolean value signifying whether or not the group has reached an MDO (this is all that is required for the naive strategy). At the other extreme, agents might need complete information about the outcome, including the action choices of each of the other agents.

Fictitious Play Learning

Fictitious play is a learning rule in which an agent assumes that each other agent is playing a fixed mixed strategy. The fictitious play agent uses counts of the actions selected by the other agents to estimate their mixed strategies and then at each round selects the action that has the highest expected value given these beliefs. Note that the fictitious play rule places very high information requirements on the agents. In order to update their beliefs, agents must have full knowledge of the outcome. Our implementation of fictitious play includes a few minor modifications to the basic rule.

One modification stems from the well known fact that agents using fictitious play may never converge to equilibrium play. Indeed our experiments show that fictitious play agents in CP DGs often generate play that oscillates within sets of outcomes, never reaching an MDO. This results from the agents' erroneous belief in the others' use of a fixed

mixed strategy. To avoid this oscillation, we modify the fictitious play rule with stochastic perturbations of agents' beliefs as suggested by (Fudenberg & Levine 1998). In particular, we apply a uniform random variation of -1% to 1% on the expected reward of each action before selecting the agent's best response.

The other modifications were necessary to make the agents' computation within each round tractable for large numbers of agents. Calculating the expected value of each possible action at each round requires time that is exponential in n . To avoid this, we store the history of play as counts of observed outcomes rather than counts of each agents' actions. Also, instead of maintaining the entire history of play, we use a bounded memory of observed outcomes. The predicted joint mixed strategy of the other agents is then calculated by assuming the observed outcomes within memory are an unbiased sample.⁵

Reinforcement Learning

Reinforcement learning is a learning rule in which agents learn a mapping from states to actions (Kaelbling, Littman, & Moore 1996). We implemented the *Q-learning* algorithm with a Boltzman exploration policy. In Q-learning, agents learn the expected reward of performing an action in a given state. Our implementation of Q-learning includes a few minor modifications to the basic algorithm.

It is well known that the performance of Q-learning is extremely sensitive to a number of implementation details. First, the choice of a state space for the agent's Q-function is critical. We chose to use only a single state, so that in effect agents learn Q-values over actions only. Second, the selection of initial Q-values and temperature is critical. We found it best to set the initial Q-values to lie strictly within the range of the highest possible payoff (i.e., being alone) and the next highest (i.e., being with one other agent). We chose to parameterize the Boltzman learning function with an initial low temperature. These choices allow agents that initially choose a non-conflicting action to have high probability of continuing to play this action, and allow those that have collided with other agents to learn eventually the true value of the action and successively choose other actions until they find an action that does not conflict.

In our implementation we chose to give the agents a selfish reward instead of the global common-preference reward. The reward is a function of the number of other agents that choose the same action, not of the degree of dispersion of the group as a whole. This selfish reward has the advantage of giving the agents a signal that is more closely tied to the effects of their actions, while still being maximal for each agent when the agents have reached an MDO.

Specialized Strategies

The first specialized strategy that we propose is the *freeze strategy*. In the freeze strategy, an agent chooses actions

⁵The reader might be concerned that this approximation changes the convergence properties of the rule. Although this may be the case in some settings, in our experiments with small n no difference was observed from those using the full history.

randomly until the first time she is alone, at which point she continues to replay that action indefinitely, regardless of whether other agents choose the same action. It is easy to see that this strategy is guaranteed to converge in the limit, and that if it converges it will converge to an MDO. The freeze strategy also has the benefit of imposing very minimal information requirements: it requires an agent to know only how many agents chose the same action as she did in the previous round.

An improvement on the freeze strategy is the *basic simple strategy*, which was originally suggested by Alpern (2001). In this strategy, each agent begins by randomly choosing an action. Then, if no other agent chose the same action, she chooses the same action in the next round. Otherwise, she randomizes over the set of actions that were either unoccupied or selected by two or more agents. Note that the basic simple strategy requires that agents know only which actions had a single agent in them after each round.

Definition 8 (Basic Simple Strategy) Given an outcome $o \in O$, an agent using the basic simple strategy will

- If $n_a^o = 1$, select action a with probability 1,
- Otherwise, select an action from the uniform distribution over actions $a' \in A$ for which $n_{a'}^o \neq 1$.

We have extended the basic simple strategy to work in the broader class of games for which $n \neq k$.

Definition 9 (Extended Simple Strategy) Given an outcome $o \in O$, an agent using the extended simple strategy will

- If $n_a^o \leq \lfloor n/k \rfloor$, select action a with probability 1,
- Otherwise, select action a with probability $\frac{n/k}{n_a^o}$ and with probability $(1 - \frac{n/k}{n_a^o})$ randomize over the actions a' for which $n_{a'}^o < \lfloor n/k \rfloor$.

Unlike the basic strategy, the extended strategy does not assign uniform probabilities to all actions that were not chosen by the correct number of agents. Consider agents reacting to the outcome $\{2, 2, 0, 0\}$. In this case each agent is better off staying with probability 0.5 and jumping to each of the empty slots with probability 0.25, than randomizing uniformly over all four slots. The extended simple strategy can actually be further improved by assigning non-uniform probabilities to the actions a' for which $n_{a'}^o < \lfloor n/k \rfloor$. We have found empirically that the learning rule converges more rapidly when agents place more probability on the actions that have fewer other agents in them. Note that the extended simple strategy requires that agents know the number of agents selecting each action in the round; the identity of these agents is not required, however.

Experimental Results

The learning rules and strategies described above differ significantly in the empirical time to converge. In Figure 2 we plot as a function of n the convergence time of the learning rules in repeated symmetric weak DGs, averaged over 1000 trials. Table 1 summarizes the observed performance of each strategy (as well as the information requirements of

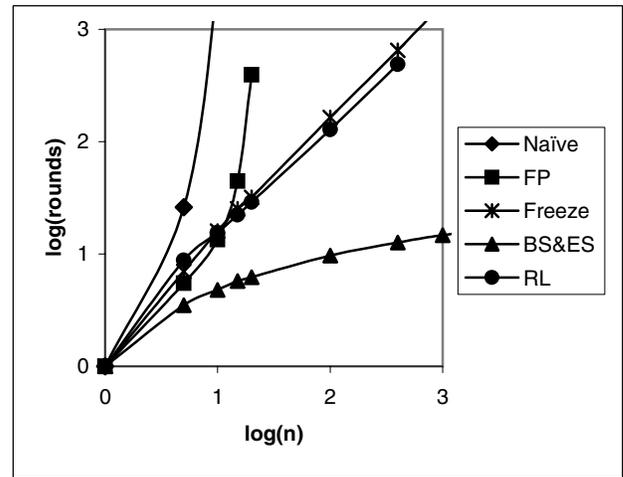


Figure 2: Log-log plot of the empirical performance of different strategies in symmetric CACP dispersion games.

Learning Rule	Information Requirements	Avg. Rounds to Converge ($f(n)$)
Naive	Whether MDO	EXP
FP	Full Information	EXP
RL	Num. in Own Action	POLY
Freeze	Num. in Own Action	LINEAR
BS & ES	Num. in All Actions	LOG

Table 1: Applicability of strategies to various classes of games with information requirements and estimated complexity class.

each strategy). We discuss the performance of each of the strategies in turn.

We begin with the learning rules. In our empirical tests we found that stochastic fictitious play always converged to an MDO. However, the number of rounds to converge was on average exponential in n . In our empirical tests of the reinforcement learning strategy we found that on average play converges to an MDO in a number of rounds that is linear in n . An interesting result is that for $n \neq k$, the algorithm didn't converge to a unique selection of actions for each agent, but rapidly adopted a set of mixed strategies for the agents resulting in average payoffs close to the optimal deterministic policy.

The specialized strategies generally exhibited better performance than the learning rules. Our empirical observations show that the number of rounds it takes for the freeze strategy to converge to an MDO is linear in n . Our empirical tests of both basic and extended simple strategies show that on average, play converges to an MDO in a number of steps that is logarithmic in the number of agents.⁶

⁶For $n > k$ certain ratios of n/k led consistently to superlogarithmic performance; slight modifications of the extended simple strategy were able to achieve logarithmic performance.

Discussion

In this paper we have introduced the class of DGs and defined several important subclasses that display interesting properties. We then investigated certain representative learning rules and tested their empirical behavior in DGs. In the future, we intend to continue this research in two primary directions.

First, we would like to further investigate some new types of DGs. We gave examples above of two classes of non-CP dispersion games that model common problems, but due to space limitations we were not able to define and characterize them in this paper. On a different note, we are also interested in a possible generalization of DGs which models the allocation of some quantity associated with the agents, such as skill or usage, to the different actions. We would like to define these classes of games formally, and explore learning rules that can solve them efficiently.

Second, we would like to continue the research on learning in DGs that we have begun in this paper. The learning rules we evaluated above are an initial exploration, and clearly many other learning techniques also deserve consideration. Additionally, we would like to complement the empirical work presented here with some analytical results. As a preliminary result, we can prove the following loose upper bound on the expected convergence time of the basic simple strategy.

Proposition 1 *In a repeated fully symmetric weak dispersion game with n agents and actions, in which all agents use the basic simple strategy, the expected number of rounds until convergence to an MDO is in $O(n)$.*

Informally, the proof is as follows. The probability that a particular agent chooses an action alone is $((n-1)/n)^{n-1}$, and so the expected number of rounds until she is alone is just $(n/(n-1))^{n-1}$. Because of the linearity of expectation, the expected number of rounds for all agents to find themselves alone must be no more than $n^n/(n-1)^{n-1}$, which is less than ne for all $n > 1$. Using similar techniques it is possible to show a quadratic bound on the expected convergence time of the freeze strategy.

Unfortunately, our empirical results show that the basic simple strategy converges in time that is logarithmic in n , and that the freeze strategy converges in linear time. This gap between our preliminary analysis and our empirical results begs future analytical work. Is it possible to show a tighter upper bound, for these learning rules or for others? Can we show a lower bound?

We would also like to better understand the optimality of learning rules. It is possible in principle to derive the optimal reactive learning rule for any finite number of agents using dynamic programming. Note that the optimal strategies obtained using this method are arbitrarily complex, however. For example, even upon reaching the simple outcome $\{2, 2, 0, 0\}$, an optimal reactive strategy for each agent chooses the same action with probability 0.5118 (not 0.5, as the extended simple strategy would dictate).

Dispersion games clearly play an important role in cooperative multiagent systems, and deserve much more discussion and scrutiny. We view the results of this paper as open-

ing the door to substantial additional work on this exciting class of games.

References

- Alpern, S. 2001. Spatial dispersion as a dynamic coordination problem. Technical report, The London School of Economics.
- Arthur, B. 1994. Inductive reasoning and bounded rationality. *American Economic Association Papers* 84:406–411.
- Azar, Y.; Broder, A. Z.; Karlin, A. R.; and Upfal, E. 2000. Balanced allocations. *SIAM Journal on Computing* 29(1):180–200.
- Balch, T. 1998. Behavioral diversity in learning robot teams.
- Brafman, R. I., and Tennenholtz, M. 2000. A near-optimal polynomial time algorithm for learning in certain classes of stochastic games. *Artificial Intelligence* 121(1-2):31–47.
- Brown, G. 1951. Iterative solution of games by fictitious play. In *Activity Analysis of Production and Allocation*. New York: John Wiley and Sons.
- Challet, D., and Zhang, Y. 1997. Emergence of cooperation and organization in an evolutionary game. *Physica A* 246:407.
- Claus, C., and Boutilier, C. 1998. The dynamics of reinforcement learning in cooperative multiagent systems. In *AAAI/IAAI*, 746–752.
- Fudenberg, D., and Levine, D. K. 1998. *The Theory of Learning in Games*. Cambridge, MA: MIT Press.
- Kaelbling, L. P.; Littman, M. L.; and Moore, A. P. 1996. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research* 4:237–285.
- Kalai, E., and Lehrer, E. 1993. Rational learning leads to nash equilibrium. *Econometrica* 61(5):1019–1045.
- Littman, M. L. 1994. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning (ML-94)*, 157–163. New Brunswick, NJ: Morgan Kaufmann.
- Osborne, M., and Rubinstein, A. 1994. *A Course in Game Theory*. Cambridge, Massachusetts: MIT Press.
- Robinson, J. 1951. An iterative method of solving a game. *Annals of Mathematics* 54:298–301.
- Schelling, T. 1960. *The Strategy of Conflict*. Cambridge, Massachusetts: Harvard University Press.