Incremental Policy Learning: An Equilibrium Selection Algorithm for Reinforcement Learning Agents with Common Interests

Nancy Fulda and Dan Ventura
Department of Computer Science
Brigham Young University
Provo, Utah, 84602
email: fulda@byu.edu, ventura@cs.byu.edu

***Abstract*— We present an equilibrium selection algorithm for reinforcement learning agents that incrementally adjusts the probability of executing each action based on the desirability of the outcome obtained in the last time step. The algorithm assumes that at least one coordination equilibrium exists and requires that the agents have a heuristic for determining whether or not the equilibrium was obtained. In deterministic environments with one or more strict coordination equilibria, the algorithm will learn to play an optimal equilibrium as long as the heuristic is accurate. Empirical data demonstrate that the algorithm is also effective in stochastic environments and is able to learn good joint policies when the heuristic's parameters are estimated during learning, rather than known in advance.**

I. Introduction

Learning to play an optimal equilibrium is a non-trivial task. Non-communicating agents must both determine the location of equilibria in the joint action space and learn which equilibria are enabled by the other agents' strategies. Wang and Sandholm describe this task in terms of two interrelated learning problems: identifying the game and learning to play [9].

We use the phrase *agents with common interests* to describe agents whose preferences coincide in at least one point in the joint action space; that is, there is at least one joint action that maximizes expected reward for all agents. We call such a joint action a *coordination equilibrium*. A coordination equilibrium is *strict* if the system contains no joint actions that maximize expected reward for one agent without maximizing it for all other players. Coordination equilibria are a special case of *optimal equilibria*, which are defined as pareto-efficient Nash equilibria. An optimal equilibrium does not necessarily maximize payoff for all players. A coordination equilibrium does.

Recent algorithms that address equilibrium selection in multiagent reinforcement learning systems include Claus and Boutilier's Joint Action Learners [1], Hu and Wellman's Nash Q-learning algorithm [3], Michael Littman's Friend-or-Foe Q-learning [6], and Wang and Sandholm's Optimal Adaptive Learning [9].

The algorithms described above share two characteristics in common. (1) They all rely on global perception of the joint action space (i.e. each agent can perceive the actions executed by its counterparts) and (2) they all make assumptions about the motives of the other agents. These two characteristics correspond to Wang and Sandholm's taxonomy: Perception of other agents' actions allows the game structure to be identified, while assumptions about other agents' motives help in determining which potential equilibria the other agent is willing to play.

Incremental Policy Learning (IPL) is a novel approach to selecting between multiple coordination equilibria. This approach uses a weakened form of condition (1) above: The agents do not need to know the entire structure of the joint action space. Instead, they require a heuristic (for example, the reward associated with a coordination equilibrium) that indicates whether the desired equilibrium was obtained. The heuristic information can be inferred from the joint action table if it is available, may be provided by an external oracle, or can be estimated based on known characteristics about the environment and observed individual rewards. One of IPL's greatest advantages is that, depending on how the heuristic information is obtained, optimal solutions may be found without learning the complete game structure.

In its most basic form, Incremental Policy Learning relies on condition (2) by assuming that a coordination equilibrium exists and by using the reward associated with this equilibrium as a parameter for the heuristic. We show in Section IV-C that this basic form can be augmented through the use of other heuristics so that the algorithm can play optimally under other assumptions about opponent motivations, including Nash-seekers and Minimax players.

II. Related Work

One of the simplest reinforcement learning techniques for selecting between multiple equilibria is to use a pre-arranged coordination mechanism such as that employed by Lauer and Riedmiller, in which the agents retain as their optimal policy the first action that successfully maximized reward [5]. This approach is effective in deterministic environments, but ceases to be effective when nondeterministic rewards are introduced or when the environment is unpredictable.

The equilibrium selection problem can also be addressed through on-policy learning. The underlying principle is that each agent's individual utilities shift to reflect the frequency with which the agent has achieved a desirable reward, causing the agents to settle towards complementary policies in which both agents benefit [1]. Unfortunately, agents using this technique do not always settle to an optimal equilibrium. Some environments particularly those resembling penalty games) camoflauge desired equilibria so that the weighted sum of received rewards still appears less desirable than some other action.

Biased exploration techniques have been utilized to encourage convergence to an optimal equilibrium. For example, Kapetanakis and Kudenko's FQM heuristic biases exploration based on the maximum reward received for a given action and the frequency with which that reward has been observed [4]. This approach increases the likelihood of convergence to an optimal equilibrium in cooperative games, but does not guarantee it.

Another option is to allow the agents to percieve the actions selected by all other agents. However, this augmentation destroys the power of on-policy learning as an equilibrium selection technique. Unlike typical reinforcement learners, whose utility estimates change in response to the biased exploration of their counterparts, agents who perceive the joint action space learn the same joint utilities regardless of the behavior of the other agents. The frequency with which each joint utility is updated is affected, but not the value to which the utilities converge. To remedy this problem, researchers have again resorted to biased-exploration techniques such as Fictitious Play and Optimistic Boltzmann [1]. This results in improved performance but does not guarantee convergence to an optimal solution.

Incremental Policy Learning resembles the biased exploration techniques described above, with the critical distinction that IPL uses information about the optimal equilibrium to guide the search. This results in guaranteed convergence to an optimal joint policy if at least one strict coordination equilibrium exists.

III. Incremental Policy Learning

A. Terminology

Let $A = \{a_1, ..., a_n\}$ be the action selections available to an agent, and let $P = \{p(a_1), ..., p(a_n)\}$ be a probability distribution over those actions Then $0 \leq p(a_i) \leq 1$ and $\sum_i p(a_i) = 1$.

We assume that the agents are repeatedly playing a single-stage game in which the (external) state of the agent never changes. In any time step t, each reinforcement learning agent executes an action $a(t) \in A$ and receives an information vector $I(t)$ back from the environment. This information includes the agent's reward $r(t)$ and may also include the action selections of other agents, the payoffs received by other agents, or other information about the environment.

The IPL algorithm uses a binary heuristic $H(t)$ that returns a value of 0 or 1. This heuristic may use as parameters the information vector $I(t)$, the agent's utilities (if it is maintaining them), or any other information accessible to the agent.

One of the simplest heuristics a reinforcement learning agent can use returns 1 if and only if the reward received by the agent matches or exceeds the expected reward associated with a coordination equilibrium, that is, if $r(t) \geq r_{max}$. An inequality is used in addition to the equality because the maximum expected reward may be exceeded due noisy payoffs. This heuristic is used for the basic form of IPL because of its simple nature and because of the potential for inferring r_{max} from observations or from *a priori* information about the environment.

B. Algorithm

The basic principle of IPL is that, if the heuristic indicates that the agent's objectives were successfully obtained, then the agent increases the probability of repeating the action just executed while slightly decreasing the probability of executing any other action. This algorithm is purely policy-based; the agent's utilities have no influence on the agent's behavior unless they are used by the heuristic function.

The basic form Incremental Policy Learning Algorithm functions as follows:

- **Initialization**
  $\forall i, p(a_i) = v_i$, where v is an arbitrarily chosen initialization vector that satisfies $\sum_{i=0}^{n} v_i = 1$ and $\forall i, v_i > 0$.

- **Action Selection**
  In each time step t, the agent selects an action $a(t) \in A$ such that $prob(a(t) = a_i) = p(a_i)$. The agent executes this action, receives an information vector $I(t)$, and updates P as described below.

- **Probability Updates**
  Let $H(t) = 1$ if $r(t) \geq r_{max}$, 0 otherwise.
  Let $0 < \alpha \leq 1$.
  If $H(t) = 1$ then $\forall i$:
  if $(a(t) = a_i)$ then $p'(a_i) = p(a_i) + \alpha(1 - p(a_i))$
  if $(a(t) \neq a_i)$ then $p'(a_i) = p(a_i) - \alpha p(a_i)$

The update rule used preserves the probability distribution P. Hence $\sum_i p'(a_i) = 1$ and for all $p'(a_i)$, $0 \leq p'(a_i) \leq 1$.

The general form of the algorithm differs from the basic form in that the heuristic $H(t)$ is not specified. It is intended that the general algorithm can be adapted to suit varied situations by selecting an appropriate heuristic.

C. Convergence

Figure 1 shows a generalized payoff matrix for a determinstic two player game. We will assume that the game is constrained such that it contains at least one coordination equilibrium and that all coordination equilibria are strict. This means that there exist values x^* and y^* such that $\forall i, j, x^* \geq x_{ij}, y^* \geq y_{ij}$, and $x_{ij} = x^*$ iff $y_{ij} = y^*$.

	b_1	b_2	...	b_m
a_1	(x_{11}, y_{11})	(x_{21}, y_{21})	...	(x_{m1}, y_{m1})
a_2	(x_{12}, y_{12})	(x_{22}, y_{22})	...	(x_{m2}, y_{m2})
...	...	...	...	...
a_n	(x_{1n}, y_{1n})	(x_{2n}, y_{2n})	...	(x_{mn}, y_{mn})

Fig. 1. Generalized payoff matrix for a two-player game

Suppose that we have two IPL agents, a and b, repeatedly playing a single-stage game with this payoff matrix using x^* and y^* as their respective r_{max} values. Under these conditions, the agents' heuristics will always be positively correlated. Hence, if (a_i, b_j) is a coordination equilibrium, then $x_{ij} = x^*$, $y_{ij} = y^*$, and $H(t) = 1$ for both agents. If (a_i, b_j) is not a coordination equilibrium, then $x_{ij} < x^*$, $y_{ij} < y^*$, and $H(t) = 0$ for both agents.

We wish to determine whether the agents will learn to play an optimal equilibrium. We begin by noting that the combination of the agents' IPL probability distributions creates a probability distribution over the joint action space such that

$$p(a_i, b_j) = p(a_i)p(b_j) \tag{1}$$

Each time a coordination equilibrium is played, the heuristic of both agents is satisfied and each joint action takes on a new probability

$$p'(a_i, b_j) = p(a_i, b_j) + \Delta p(a_i, b_j) \tag{2}$$

The value to which $p(a_i, b_j)$ can increase is bounded for all joint actions that are not coordination equilibria, as shown in Theorem 3.1.

Theorem 3.1: If (a_i, b_j) is not a coordination equilibrium and if $p(a_i, b_j) > 0.5$, then $\Delta p(a_i, b_j) \leq 0$.

Proof: We wish to establish the conditions under which $\Delta p(a_i, b_j) \leq 0$. Using equation (2) to derive $\Delta p(a_i, b_j)$ and substituting, we obtain $p'(a_i, b_j) - p(a_i, b_j) \leq 0$, which by equation (1) is equivalent to

$$p'(a_i)p'(b_j) - p(a_i)p(b_j) \leq 0 \tag{3}$$

The values of $p'(a_i)$ and $p'(b_j)$ depend on the nature of the action executed by the system at time t. Let (a_k, b_z) be this joint action. If (a_k, b_z) is not a coordination equilibrium, then the IPL probabilities are not adjusted. In this case, $p'(a_i) = p(a_i)$, $p'(b_j) = p(b_j)$ and $\Delta(a_i, b_j) = 0$.

If (a_k, b_z) is a coordination equilibrium then joint action (a_i, b_j) can be related to it in three ways: They can share a row $(i = k, j \neq z)$, share a column $(i \neq k, j = z)$, or be completely disjoint $(i \neq k, j \neq z)$. They cannot share both a row and a column because (a_k, b_z) is a coordination equilibrium and (a_i, b_j) is not.

If the two actions are disjoint then $p'(a_i) = p(a_i) - \alpha p(a_i)$ and $p'(b_j) = p(b_j) - \alpha p(b_j)$. This means that $p'(a_i, b_j) < p(a_i, b_j)$, so $\Delta p(a_i, b_j) < 0$.

If the actions share a row or a column the situation is less clear because one term of $p'(a_i)p'(b_j)$ is increasing while the other is decreasing. Without loss of generality, assume $(i = k, j \neq z)$. Then $p'(a_i) = p(a_i) + \alpha(1 - p(a_i))$ and $p'(b_j) = p(b_j) - \alpha p(b_j)$.

Substituting into equation (3) we get

$$(p(a_i) + \alpha(1 - p(a_i)))(p(b_j) - \alpha p(b_j)) - p(a_i)p(b_j) \leq 0 \tag{4}$$

which resolves to

$$p(a_i)(\alpha - 2) \leq (\alpha - 1) \tag{5}$$

Dividing by $\alpha - 2$, a negative quantity, we find that $\Delta p(a_i, b_j) \leq 0$ whenever $p(a_i) > \frac{\alpha - 1}{\alpha - 2}$. The right side of the equation is maximized when $\alpha = 0$, so as long as $p(a_i) > 0.5$, $\Delta(a_i, b_j) \leq 0$.

■

The above proof demonstrates that the system can never converge to a solution that is not a coordination equilibrium: whenever the probability of a suboptimal joint action is greater than 0.5, it will decrease on the next update. We also notice an interesting phenomenon. In equation 4, the value $p(b_j)$ exists in all multiplicative terms and factors out of the calculation. Thus, if a suboptimal joint action (a_i, b_j) shares a column with a coordination equilibrium, the sign of $\Delta p(a_i, b_j)$ depends only on the value of $p(a_i)$.

Let us now consider the case where (a_i, b_j) is a coordination equilibrium. In this case we find that there is some critical value of $p(a_i, b_j)$ beyond which $\Delta p(a_i, b_j)$ is always positive.

Theorem 3.2: If a system contains at least one strict coordination equilibrium and at least one joint action that is not a coordination equilibrium, then $\exists p^*$ such that if $p(a_i, b_j) > p^*$ then $p(a_i, b_j)$ is more likely to increase over time than to decrease.

Proof: We wish to determine under what conditions $\Delta p(a_i, b_j) > 0$. In order to do so, we examine two types of update situations. Since (a_i, b_j) is a coordination equilibrium, $p(a_i)$ and $p(b_j)$ will increase each time it is executed. When some other coordination equilibrium $(a_{k \neq i}, b_{z \neq j})$ is executed, $p(a_i)$ and $p(b_j)$ both decrease. To reflect these situations, we define

$$p^+(a_i, b_j) = (p(a_i) + \alpha(1 - p(a_i)))(p(b_j) + \alpha(1 - p(b_j))) \tag{6}$$

$$p^-(a_i, b_j) = (p(a_i) - \alpha p(a_i))(p(b_j) - \alpha p(b_j)) \tag{7}$$

Let $p(c)$ represent the probability that the executed action is some coordination equilibrium other than (a_i, b_j). Then the value of $p'(a_i, b_j)$ can be probabilistically described as a weighted average of the above possibilities, using $p(a_i, b_j)$ and $p(c)$ as weighting factors. When $p'(a_i, b_j) - p(a_i, b_j) > 0$, $\Delta p(a_i, b_j)$ is positive. We therefore seek the conditions that satisfy

$$\frac{p(a_i, b_j)p^+(a_i, b_j) + p(c)p^-(a_i, b_j)}{p(a_i, b_j) + p(c)} - p(a_i)p(b_j) > 0 \tag{8}$$

We note that the term $p(c)p^{-}(a_i, b_j)$ is somewhat pessimistic, since the executed coordination equilibrium might share a row or column with (a_i, b_j). In this case, either $p(a_i)$ or $p(b_j)$ would increase, rather than decrease. However, these cases only improve the chance that the inequality in equation (8) will be satisfied. We therefore assume the worst-case scenario that the executed coordination equilibrium shares neither a row nor a column with (a_i, b_j).

After substituting from equations (6) and (7), equation (8) can be reduced to

$$p(a_i)p(b_j) + p(c) < \frac{\alpha + (1-\alpha)(p(a_i) + p(b_j))}{2-\alpha} \quad (9)$$

In the worst-case scenario, $p(c) = 1 - p(a_i)p(b_j)$. In this case, equation (9) can be simplified to $p(a_i) + p(b_j) > 2$; an impossible condition. However, $p(c) = 1 - p(a_i)p(b_j)$ only if all joint actions are coordination equilibria, which defies the premises of the theorem. We quickly see that for any value of $p(c) = 1 - p(a_i)p(b_j) - \epsilon$, where $0 < \epsilon < 1 - p(a_i)p(b_j)$, then equation (9) resolves to

$$p(a_i) + p(b_j) > 2 - \frac{\epsilon(2-\alpha)}{1-\alpha} \quad (10)$$

which is not an impossible condition. There are values of $p(a_i)$ and $p(b_j)$ that will satisfy it, even though they may be high. Thus there exist critical values for $p(a_i)$ and $p(b_j)$ (and correspondingly, for $p(a_i, b_j)$, beyond which $\Delta p(a_i, b_j)$ tends to be positive.

■

We can now proceed to examine the overall system behavior. By Theorem 3.2, there exists some critical value p^* beyond which $p(a_i, b_j)$ tends to be continually increasing. The greater the amount by which $p(a_i, b_j)$ exceeds this threshhold, the less likely it becomes that $p(a_i, b_j)$ will decrease. The threshhold effectively represents the point at which a single coordination equilibrium dominates all other possibilities and begins to be executed with steadily increasing frequency.

With continued positive updates, $p(a_i)$ and $p(b_j)$ converge towards 1 (because the iterative series $z' = z + \alpha(1 - z)$ converges to 1 for $0 < \alpha \leq 1$). Consequently, $p(a_i, b_j)$ also converges to 1.

Getting the ball rolling on this convergence issue may take a while. If p^* is relatively high, then it may take many iterations before the probability of one of the coordination equilibria happens to reach it. Fortunately, we have a guarantee from Theorem 3.1 that no suboptimal equilibrium can maintain a probability greater than 0.5, so there is no risk of converging suboptimally while waiting for a coordination equilibrium to dominate. In the degenerate case where all joint actions are optimal, the agents will, of course, also play optimally, even if their probabilities never converge.

IV. Algorithm Behavior

Most environments are not conveniently deterministic, and most heuristics are not 100% accurate. How does the IPL algorithm perform in the face of such uncertainties?

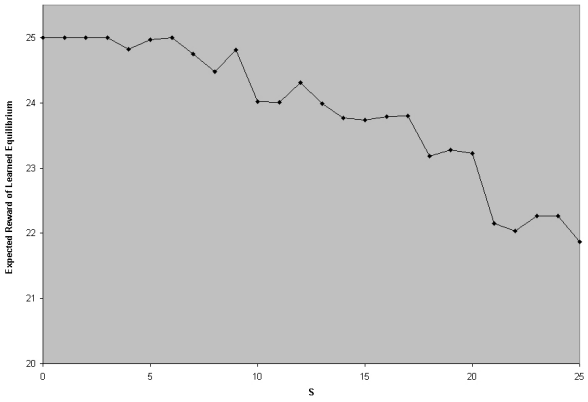

Fig. 2. IPL performance as a function of noise

A. Noisy Rewards

Figure 2 shows algorithm performance as a function of noise. Two IPL agents, each having five actions, repeatedly played a single-stage game until they executed the same joint action 50 times in succession. Each cell of the payoff matrix was randomly initialized to an integer between 0 and 24 (different random payoffs were assigned to each agent), with the exception of 5 randomly placed coordination equilibria whose payoff was 25 for both agents.

Gaussian noise was simulated using Peitgen et. al's equation [2]:

$$D = \frac{1}{A}\sqrt{\frac{12}{n}}\sum_{i=1}^{n} Y_i - \sqrt{3n} \quad (11)$$

Using values $A = 100$ and $n = 50$. The result D was multipled by a scaling factor of S to provide different degrees of noise. The value SD was added to the agents' reward signals.

The agents used $r_{max} = 25$ and $\alpha = 0.1$, and the results of 100 trials were averaged for each data point.

We found that even for very large values of S with respect to the range of possible rewards, the IPL algorithm was able to focus in on a near-optimal solution with reasonable frequency.

B. Estimation of r_{max}

A critical question for IPL is how the algorithm performs when the value of r_{max} is not known in advance, but must instead be inferred from known information.

We implemented a set of Q-learning agents who were each able to observe the action selections of the other agents. The agents used an initial utility value of 0, a learning rate $\eta = 0.1$, and a simplified joint Q-value update equation

$$Q'(a_i, b_j) = (1-\eta)Q(a_i, b_j) + \eta(r) \quad (12)$$

where $Q(a_i, b_j)$ is the estimated utility of performing joint action (a_i, b_j) and r is the reward received for executing joint action (a_i, b_j).

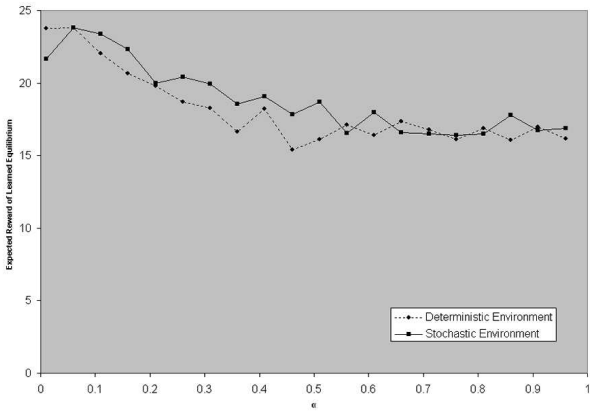

Fig. 3. IPL performance as a function of α with r_{max} estimation. Scaling factor $S = 0$ was used for the deterministic environment, $S = 5$ for the stochastic environment.

In each time step, each agent estimated $r_{max} = max_{i,j}\ Q(a_i, b_j)$ and exceuted the IPL update equation. Payoff matrix generation and noise generation were carried out as described in the previous sub-section, and the results of 100 trials were averaged.

The results are shown in Figure 3. As one might expect, IPL performance was better with lower values of α. This is not surprising, since a high value for α might cause the IPL algorithm to converge before the joint Q-values accurately reflected the relative magnitude of the expected rewards.

C. Extensions of the Algorithm

The IPL algorithm lends itself naturally to several possible heuristics. Here, we discuss only a few of those which intuitively appear most useful and which seem to reflect other results obtained in the field of multiagent reinforcement learning.

If agents have access to the full payoff matrix of the game being played (either because it was provided *a priori* or because they have inferred it by watching the action selections and rewards of other actions), then they can determine whether a given joint action represents a Nash equilibrium. By letting $H(t) = 1$ if the executed action was a Nash equilibrium and $H(t) = 0$ otherwise, the agents can select between multiple Nash equilibria.

Similarly, agents interacting in strictly adversarial environments can learn to play a Minimax strategy byletting $H(t) = 1$ if the executed joint action is a Minimax solution and $H(t) = 0$ otherwise. Naturally, both of these approaches require that the agents be able to see the action selections of their counterparts, and also rely on the assumption that all agents in the system are using the same heuristic.

One potentially interesting application of the IPL algorithm is in satisficing environments. In satisficing, agents seek actions that are "good enough" rather than seeking actions that are optimal [7], [8]. A satisficing criterion could be chosen (such as a threshhold value for r_{max}) such that $H(t) = 1$ whenever the criterion is satisfied. In this way, the agents could select between multiple satisficing solutions.

V. Conclusions and Future Work

We have presented an equilibrium selection method for agents that are able to determine with reasonable accuracy whether or not an optimal equilibrium was obtained in the last time step. When heuristic information is provided by an external source, the algorithm is able to search through policy space directly, without learning utilities or observing the actions of other agents. When the heuristic information must be inferred or approximated based on observed rewards, then utilities and observation of the complete action space may be useful tools for the acquisition of good heuristic data.

Incremental Policy Learning provably learns to play an optimal solution if the heuristic is accurate, the environment is deterministic, and at least one coordination equilibrium exists.

Empirical studies indicate that IPL performs well in the presence of noise and when the heuristic information is approximated rather than precisely known. Future work should concentrate on theoretical bounds on the effectiveness of such methods, as well as on an analysis of the effect of the joint action space size on convergence speed.

Because IPL updates only when the heuristic's requirements are satisfied, convergence may take a very long time in large joint action spaces with sparse equilibria. Compounding this problem, if the environment is noisy and the joint action space has several near-optimal solutions, the agents may learn to play one of these before a true optimum is discovered. Both of these problems might be alleviated by adding a decay factor such that the probability of an action's execution decreases each time an optimal solution is not found. This would encourage more uniform exploration of the joint action space.

Finally, other possible heuristics and methods for approximating them should be developed so that the algorithm's usefulness can be expanded to new situations.

References

[1] Caroline Claus and Craig Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *AAAI/IAAI*, pages 746–752, 1998.

[2] H. Jurgens H. O. Peitgen and D. Supe. *Chaos and Fractals*. Springer-Verlag, New York, 1992.

[3] J. Hu and M. Wellman. Nash q-learning for general-sum stochastic games. *Journal of Machine Learning Research*, to appear, 2003.

[4] S. Kapetanakis and D. Kudenko. Improving on the reinforcement learning of coordination in cooperative multi-agent systems. In *Second AISB Symposium on Adaptive Agents and Multi-Agent Systems*, 2002.

[5] Martin Lauer and Martin Riedmiller. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In *Proceedings of the 17th International Conference on Machine Learning*, pages 535–542, San Francisco, CA, 2000. Morgan Kaufman.

[6] Michael Littman. Friend or foe q-learning in general-sum markov games. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 322–328, 2001.

[7] H. A. Simon. A behavioral model of rational choice.

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

[9] X. Wang and T. Sandholm. Reinforcement learning to play an optimal nash equilibrium in team markov games. In *Advances in Neural Information Processing Systems 15 (NIPS-2002)*, Vancouver, 2002.