# Convergence Problems of General-Sum Multiagent Reinforcement Learning

**Michael Bowling** MHB@CS.CMU.EDU
Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213-3890

## Abstract

Stochastic games are a generalization of MDPs to multiple agents, and can be used as a framework for investigating multiagent learning. Hu and Wellman (1998) recently proposed a multiagent Q-learning method for *general-sum* stochastic games. In addition to describing the algorithm, they provide a proof that the method will converge to a Nash equilibrium for the game under specified conditions. The convergence depends on a lemma stating that the iteration used by this method is a contraction mapping. Unfortunately the proof is incomplete. In this paper we present a counterexample and flaw to the lemma's proof. We also introduce strengthened assumptions under which the lemma holds, and examine how this affects the classes of games to which the theoretical result can be applied.

## 1. Introduction

One of the greatest difficulties of learning in a multiagent domain is that the world does not appear stationary from an agent's perspective. The effects of an agent's actions depend on the actions of the other agents, which are likely to be changing as the other agents also learn to improve their behavior. The result is that the value of an agent's policy depends upon the behavior of the other agents. In addition, the value might change even though the policy is not changing, as the other agents learn and adapt. This problem requires the traditional notion of optimality to be abandoned, since most tasks do not have optimal policies independent of the policies of the other agents.

The field of game theory specifically addresses these multiagent issues, by providing a theoretical framework for examining policy selection. Specifically, stochastic games (Shapley, 1953) offer a compelling model for multiagent learning. Stochastic games are a natural extension of traditionally single-agent Markov decision processes (MDPs) to include multiple agents. Game theory also provides the notion of a Nash equilibria, which are a set of policies for the players such that no player would do better by deviating from its policy. Minimax-Q (Littman, 1994) was one of the first reinforcement learning algorithms for stochastic games and directly learns an equilibrium policy in *zero-sum* stochastic games.

Recently, Hu and Wellman (1998) introduced a new algorithm for learning in *general-sum* stochastic games. This algorithm, like Minimax-Q, is designed to directly learn a Nash equilibrium. In addition to the algorithm, they presented a theoretical analysis proving their algorithm will converge to an equilibrium under certain conditions. Unfortunately their proof is not complete. In this paper we present a counterexample and flaw to the proof of their crucial lemma. We also introduce strengthened assumptions under which the lemma and main theorem are valid, but rather limited.

In Section 2 we present a brief overview of the stochastic game framework and the necessary results from game theory. In Section 3 we present the Hu and Wellman algorithm and outline their convergence proof. In Section 4 we introduce the counterexample and flaw in their proof, and also the strengthened assumptions under which their result is valid. In Section 5 we discuss the ramifications of this result before concluding.

## 2. Stochastic Game Framework

A *stochastic game* is a tuple $(n, \mathcal{S}, \mathcal{A}_{1\ldots n}, T, R_{1\ldots n})$, where $n$ is the number of agents, $\mathcal{S}$ is a set of states, $\mathcal{A}_i$ is the set of actions available to agent $i$ (and $\mathcal{A}$ is the joint action space $\mathcal{A}_1 \times \ldots \times \mathcal{A}_n$), $T$ is a transition function $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$, and $R_i$ is a reward function for the $i$th agent $\mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$. This looks very similar to the MDP framework except we have multiple agents selecting actions and the next state and rewards depend on the joint action of the agents. It's also important to notice that each agent has its own separate reward function. The goal for each agent is to select actions in order to maximize its discounted future rewards with discount factor $\gamma$.

Stochastic games are a very natural extension of MDPs to multiple agents. They are also an extension of matrix games to multiple states. Two example matrix games are in Table 1. In these games there are two players; one selects a row and the other selects a column of the matrix. The entry that they jointly select determines the payoffs according to their matrix. In Table 1 the matching pennies game is a zero-sum matrix game, since the column player always receives the negative of the payoff of the row player. General-sum games, of which the coordination game is an example, do not have any restriction on the players' payoffs.

*Table 1.* Matching pennies and coordination matrix games. Matching pennies is a zero-sum game, while the coordination game is general-sum.

$$R_{\text{row}} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \qquad R_{\text{col}} = \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}$$

Matching Pennies

$$R_{\text{row}} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \qquad R_{\text{col}} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

Coordination Game

Each state in a stochastic game can be viewed as a matrix game with the payoffs for each joint action determined by the matrices $R_i(s, a)$. After playing the matrix game and receiving their payoffs the players are transitioned to another state (or matrix game) determined by their joint action. We can see that stochastic games then contain both MDPs and matrix games as subsets of the framework.

**Mixed Policies.** Unlike in single-agent settings, deterministic policies in multiagent settings can often be exploited by the other agents. Consider the matching pennies matrix game as shown in Table 1. If the column player were to play either action deterministically, the row player could win (and column player lose) every time. This requires us to consider stochastic or mixed strategies and policies. A stochastic policy, $\rho : \mathcal{S} \to PD(\mathcal{A}_i)$, is a function that maps states to mixed strategies, which are probability distributions over the player's actions.

**Nash Equilibria.** Even with the concept of mixed strategies there are still no optimal strategies that are independent of the other players' strategies. We can, though, define a notion of best-response. A strategy is a *best-response* to the other players' strategies if it is optimal given their strategies. The major advancement that has driven much of the development of matrix games, game theory, and even stochastic games is the notion of a best-response equilibrium, or *Nash equilibrium* (Nash, Jr., 1950).

A Nash equilibrium is a collection of strategies for each of the players such that each player's strategy is a best-response to the other players' strategies. So, no player can do better by changing strategies given that the other players also don't change strategies. What makes the notion of equilibrium compelling is that all matrix games have such an equilibrium, possibly multiple equilibria. Zero-sum two-player games, where one player's payoffs are the negative of the other, have a *single* Nash equilibrium.[1] In the matching pennies example in Table 1, the equilibrium consists of each player playing the mixed strategy where both actions have equal probability. In the coordination game, there are two pure (or deterministic) equilibria: both players select their first action, or both players select their second action.

The concept of equilibria also extends to stochastic games. This is a non-trivial result, proven by Shapley (1953) for zero-sum stochastic games and by Filar and Vrieze (1997) for general-sum stochastic games.

**Minimax-Q.** Littman (1994) introduced a reinforcement learning technique for zero-sum games that directly learns the game's Nash equilibrium. The algorithm, Minimax-Q, extends Q-learning in order to explicitly reason about multiple agents. The algorithm maintains $Q$ values for every state/joint-action pair. The entry $Q(s, a)$ approximates the expected discounted reward if the players select joint action $a$ from state $s$ and then follow the stochastic game's Nash equilibrium. Given an observation, $\langle s, a, s', r \rangle$, consisting of a state, joint-action of the players, resulting next state, and rewards, the $Q$ values can be updated. The update rule uses a learning parameter $\alpha$ and performs the following computation,

$$Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha(r + \gamma V(s')),$$

where,

$$V(s') = \text{Value}\left[Q(s', \bar{a})\right]_{\bar{a} \in \mathcal{A}}.$$

This update rule is very similar to standard Q-learning but with the next state's value involving the computation of the value of the matrix game that corresponds to the $Q$ values for that state.[2]

Littman proved that this technique will converge to the game's Nash equilibrium with the usual Q-learning assumptions on exploration and learning rate. An interesting aspect of this result is that the $Q$ values will converge regardless of the actions selected by the opponent.

[1] There can actually be multiple equilibria, but they will all have equal payoffs and are interchangeable (Osborne & Rubinstein, 1994).

[2] The value of this zero-sum matrix game is $\max_{\sigma \in PD(\mathcal{A}_1)} \min_{a_2 \in \mathcal{A}_2} \sum_{a_1 \in \mathcal{A}_1} Q(s, \langle a_1, a_2 \rangle)\sigma_{a_1}$, which can be solved using linear programming.

## 3. Q-Learning for General-Sum Games

Hu and Wellman (1998) introduced a Q-learning method for solving general-sum stochastic games. Their algorithm is an extension of Minimax-Q, and replaces the computation of the value of a zero-sum game with the computation of the value of a general-sum game. The algorithm strives to maintain the same property of Minimax-Q that it can learn equilibria independent of the actions selected by the other players.

The algorithm, like Minimax-Q, maintains $Q$ values for every state/joint-action pair. Since the agents' rewards in general-sum games can be completely independent, multiple $Q$ values (one for each agent) must be maintained. The entry $Q^k(s, a)$ approximates the expected discounted reward for player $k$ if the players select joint action $a$ from state $s$ and then follow the same Nash equilibrium of the stochastic game. On an observation, $\langle s, a, s', r^i \rangle$ where $r^i$ is player $i$'s reward, the update is,

$$Q^i(s, a) \leftarrow (1 - \alpha)Q^i(s, a) + \alpha(r^i + \gamma V^i(s')),$$

where,

$$V^i(s') = \text{Value}^i\left[Q(s')\right]$$

For ease of notation we use $Q(s)$ to represent the general-sum matrix game that is represented by the $n$ matrices whose entries correspond to $Q^i(s, a \in \mathcal{A})$. So in the update rule the Value$^i$ operation computes the equilibrium value to player $i$ of the general-sum matrix game defined by the $Q$ values at the state, $s'$.[3] Since the update operation makes use of the equilibrium value of a state, this is only well-defined if there is a single equilibrium (or multiple equilibria with the same value.) It is assumed for their algorithm and theoretical results that there is in fact a unique equilibrium.

Their main result is the theorem below. The theorem claims convergence of the algorithm for two-player, general-sum games. In the theorem, $r_t^k$ refers to the immediate reward received by player $k$ at time $t$, and $\pi^1(s)Q^k(s)\pi^2(s)$ refers to player $k$'s expected value of playing the matrix game $Q(s)$ where the players select actions according to the strategies, $\left(\pi^1(s), \pi^2(s)\right)$.

**Theorem 1** *(Hu & Wellman, 1998) Under assumptions below, let the sequences $\left(Q_t^1, Q_t^2\right)$ be updated by,*

$$\begin{aligned} Q_{t+1}^k(s, \langle a^1, a^2 \rangle) &= (1 - \alpha_t)Q_t^k(s, \langle a^1, a^2 \rangle) + \\ &\quad \alpha_t \left(r_t^k + \gamma\pi^1(s')Q_t^k(s')\pi^2(s')\right), \end{aligned}$$

*where $\left(\pi^1(s), \pi^2(s)\right)$ is a mixed strategy Nash equilibrium for the matrix game $\left(Q_t^1(s'), Q_t^2(s')\right)$. Then these sequences converge to the Nash equilibrium Q values $\left(Q_*^1, Q_*^2\right)$ defined by,*

$$Q_*^k(s, \langle a^1, a^2 \rangle) = r_t^k + \gamma\pi_*^1(s')Q_t^k(s')\pi_*^2(s'),$$

*where $\left(\pi_*^1(s), \pi_*^2(s)\right)$ is a Nash equilibrium for the stochastic game.*

[3] Finding the value of a general-sum game requires an involved quadratic programming solution (Filar & Vrieze, 1997).

The theorem requires three assumptions. Two of these assumptions deal with exploration and the proper decay of the learning rate. These are the standard assumptions of Q-learning and are not presented in this paper. The final assumption deals with the nature of the matrix games that are faced while learning.

**Assumption 1** *(Hu & Wellman, 1998) A Nash equilibrium $\left(\pi^1(s), \pi^2(s)\right)$ for any matrix game $\left(Q_t^1(s), Q_t^2(s)\right)$ satisfies one of the following properties:*

1. *The equilibrium is a global optimal.*

$$\forall \rho^k \quad \pi^1(s)Q^k(s)\pi^2(s) \geq \rho^1(s)Q^k(s)\rho^2(s)$$

2. *The equilibrium receives a higher payoff if the other agent deviates from the equilibrium strategy.*

$$\forall \rho^k \qquad \begin{aligned} \pi^1(s)Q^1(s)\pi^2(s) &\leq \pi^1(s)Q^1(s)\rho^2(s) \\ \pi^1(s)Q^2(s)\pi^2(s) &\leq \rho^1(s)Q^2(s)\pi^2(s) \end{aligned}$$

It is interesting to note what matrix games the properties encompass. The first property states that there's a set of strategies for the players where each player individually receives its maximum possible payoff. This also ensures that such a set of strategies is an equilibrium, since no player could benefit from deviating from the strategy. The second property states that the Nash equilibrium for the game is a "saddle point". This means that not only does a player *not benefit* from deviating from the equilibrium, but also that all other players *do benefit* if the player deviates. Notice that all zero-sum games satisfy this second property.

The proof of their theorem makes use of a crucial lemma claiming their update function is a contraction mapping. Let $P_t^k$ be an update function defined as,

$$P_t^k Q^k(s) = r_t^k + \gamma\pi^1(s)Q^k(s)\pi^2(s),$$

where $(\pi^1(s), \pi^2(s))$ are a Nash equilibrium for the two-player matrix game defined by the matrices $\left(Q^1(s), Q^2(s)\right)$. Notice that this is the same update function as in Theorem 1, but without the stochastic approximation. Their lemma claims that $P_t^k$ satisfies the following property,

$$\forall Q^k \qquad ||P_t^k Q^k - P_t^k Q_*^k|| \leq \gamma||Q^k - Q_*^k||,$$

where $||\cdot||$ is the max-norm operator over all states and actions. This lemma effectively states that the update function will always move $Q^k$ closer to $Q_*^k$.

## 4. Counterexample and Flaw

In this section we show a counterexample to their lemma and also the flaw in their proof of it. We follow this by introducing strengthened assumptions under which the lemma is valid.

### 4.1 Counterexample

Consider the stochastic game with three states shown in Figure 1. State $s_0$ always transitions to state $s_1$ with rewards 0. State $s_1$ is a $2 \times 2$ game with all actions causing the game to transition to a terminating state, $s_2$.

Consider the following $Q$ function,

$$\begin{aligned} Q(s_0) &= (\gamma, \gamma) \\ Q(s_1) &= \begin{pmatrix} \boxed{1+\epsilon, 1+\epsilon} & 1-\epsilon, 1 \\ 1, 1-\epsilon & 1-2\epsilon, 1-2\epsilon \end{pmatrix} \\ Q(s_2) &= (0, 0). \end{aligned}$$

The matrix game corresponding to $Q(s_1)$ has a unique Nash equilibrium where both players choose their first action. Notice that $||Q - Q_*|| = \epsilon$. If $P$ is a contraction mapping then after applying $P$ the values should be closer to $Q_*$. The actual values for $PQ$ are,

$$\begin{aligned} PQ(s_0) &= (\gamma(1+\epsilon), \gamma(1+\epsilon)) \\ PQ(s_1) &= \begin{pmatrix} 1, 1 & 1-2\epsilon, 1+\epsilon \\ 1+\epsilon, 1-2\epsilon & \boxed{1-\epsilon, 1-\epsilon} \end{pmatrix} \\ PQ(s_2) &= (0, 0). \end{aligned}$$

Notice that $||PQ - PQ_*|| = 2\gamma\epsilon$, which is greater than $\epsilon$ if $\gamma > 0.5$. Hence, with the max norm $P$ is not a contraction mapping and the lemma is false.

### 4.2 Proof Flaw

The flaw in their proof of the lemma is due to a missing case. It handles the case where $Q_*(s)$ meets property 1 of Assumption 1, and the case where $Q(s)$ meets property 2 of Assumption 1. It fails to address when $Q_*(s)$ meets property 2 and $Q(s)$ meets property 1. This is exactly the case of the counterexample. The matrix games $Q(s_1)$ and $Q_*(s_1)$ both have unique pure strategy equilibriums (as shown by the boxes in their respective matrices.) The Nash equilibrium of $Q(s)$ meets property 1 because both players playing their first action is the best either player can do. The Nash equilibrium of $Q_*(s)$ meets property 2 because it's a "saddle point", that is if a player deviates from the equilibrium the other player gets rewarded. So the counterexample does indeed satisfy their assumptions.

It is important to note that the counterexample and flaw only show that the backup used by the algorithm is not a contraction mapping using the max norm. This does not necessarily disprove Theorem 1, as there may be another norm for which the backup *is* a contraction mapping. On the other hand, it leaves open whether the theorem is actually true and provides evidence to the contrary, which is discussed further in Section 5.2.

### 4.3 Strengthened Assumptions

The original assumption on which their proof is based can be strengthened in order to make the crucial lemma true. This is done by simply ruling out the case not handled in their proof and under which the counterexample falls.

**Assumption 2** *The Nash equilibrium of all matrix games, $Q_t(s)$, as well as $Q_*(s)$ must satisfy property 1 in Assumption 1* ***or*** *the Nash equilibrium of all matrix games, $Q_t(s)$, as well as $Q_*(s)$ must satisfy property 2 of Assumption 1.*

This new assumption, although making the original theorem true, places heavy limitations on the applicability of the results. This is examined in the next section.

## 5. Discussion

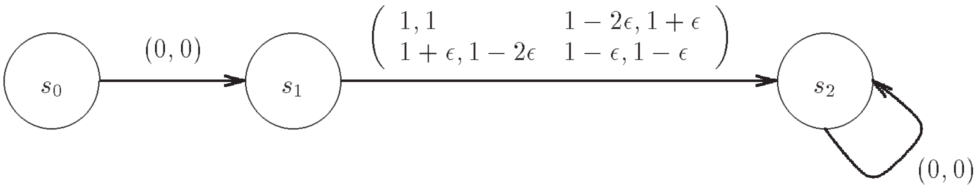
In this section we discuss the applicability of Theorem 1 with the strengthened assumption that it requires. We also examine other issues surrounding the problem of convergence in general-sum stochastic games.

### 5.1 Applicability of the Theorem

Under strengthened Assumption 2 the theorem has a number of limitations. One of these limitations was also problematic under the original assumption, but is even more so under Assumption 2.

The first limitation is that if the current $Q_t$ values satisfy the assumption and the $Q_*$ values are known to satisfy the assumption, there's still no guarantee that future $Q_{t+1}$ values will satisfy the assumption. Since no guarantees are made on future values, the theorem cannot actually guarantee convergence. Rather it only guarantees that *if* the $Q$ values converge while always satisfying the assumption, then they've converged to the game's equilibrium. Since Assumption 2 makes a further requirement that the values always satisfy the *same property*, this only magnifies this problem.

The second limitation is due directly to the strengthened assumption. The $Q$ values must always satisfy the same property that the unknown $Q_*$ values satisfy. Although, the $Q$ values can be initialized to satisfy both properties (e.g. by initializing all the values to zero), the initial steps of learning will undoubtedly move the $Q$ values into satisfying only one of the properties. The theorem, then, only




$$Q_*(s_1) = (\gamma(1-\epsilon), \gamma(1-\epsilon)) \qquad Q_*(s_1) = \begin{pmatrix} 1,1 & 1-2\epsilon, 1+\epsilon \\ 1+\epsilon, 1-2\epsilon & \boxed{1-\epsilon, 1-\epsilon} \end{pmatrix} \qquad Q_*(s_2) = (0,0)$$

*Figure 1.* A three state stochastic game, including the Nash equilibrium values, $Q_*$. All transitions are deterministic and are independent of the actions selected. The only choice available to the agents is in state $s_1$, where the corresponding matrix game has a unique Nash equilibrium where both players choose their second action.

guarantees convergence if the unknown $Q_*$ values for the stochastic game satisfies the *exact same* property.

### 5.2 Convergence in General-Sum Games

One question that arises is why general-sum games are more problematic for convergence of learning algorithms. The counterexample presented in this paper gives some insight into this question. Small changes in the values of joint-actions can cause a large change in the state's Nash equilibria. This can cause drastic changes in the value of that state. Overall, this means small changes in values can propagate into even larger changes in values. This is evident in the counterexample where an $\epsilon$ change in the $Q$ values propagates into a $2\epsilon$ change.

Despite this fact, there are some classes of general-sum stochastic games where convergent learning is not so problematic. In fact, there is a large class where even naive single-agent learners, such as Q-learning (Watkins, 1989), can find Nash equilibria. Fully collaborative games, where all the agents have identical reward functions, can be learned by simple Q-learners (Claus & Boutilier, 1998). Games that are iterated dominance solvable, where the process of eliminating universally inferior policies by the players leaves only Nash equilibria policies, are also solvable by naive learning (Fudenberg & Levine, 1999).

In fact even stochastic games that do not fall under these two categories can still be solved with single-agent learners. Consider the two-agent gridworld game depicted in Figure 2, which was introduced as an example domain for the Hu and Wellman algorithm (Hu, 1999). The agents start in two corners and are trying to reach the goal square on the opposite wall. The players have the four compass actions (i.e. N, S, E, and W), which are in most cases deterministic. If the two players attempt to move to the same square, both moves fail. To make the game interesting and force the players to interact, from the initial starting position the North action is uncertain, and is only executed with probability 0.5. The optimal path for each agent, then, is to move laterally on the first move and then move North to the goal, but if both players move laterally then the actions will fail. There are two Nash equilibria for this game. They involve one player taking the lateral move and the other trying to move North.

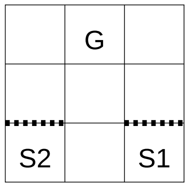



*Figure 2.* Gridworld game. The dashed walls represent the actions that are uncertain.

*Table 2.* Q-learning for the gridworld game. The table shows the number of trials where the players converged to a particular policy. "W-N" and "E-N" correspond to the two Nash equilibria where one player moves laterally in initial state while the other tries to move North. "Other" corresponds to all other policies.

| Strategy | Trials | |
|---|---|---|
| W-N | 114 | (57%) |
| N-E | 86 | (43%) |
| Other | 0 | (0%) |
| Total | 200 | (100%) |

Table 2 shows the results of training two agents in the gridworld game, using standard Q-learning. The agents were trained with a decayed learning rate for one million steps to be sure the policies had converged. The table shows how often the agents converged to specific policies in the 200 trials. The important thing to note is that the agents always converged to one of the game's Nash equilibria. So naive single-agent learners can also converge to Nash equilibrium in games that aren't fully-collaborative or iterated dominance solvable. Understanding what characterizes these easier general-sum games may be helpful for isolating the difficulties present in other general-sum games.

### 5.3 Static Solutions

It is important to note that there do exist static algorithms for solving general-sum stochastic games. An equilibrium solution can be constructed by solving a set of nonlinear complementarity problems, one for each state (Filar & Vrieze, 1997). This implies a model-based learning algorithm could be constructed, where the transition and reward functions are learned through experience, but the policy is constructed using the static solver. Not only is this likely to be computationally intractable, it's also unknown whether it might encounter the same "convergence" problems. In particular, a small error in the learned transition probabilities and rewards might cause a large difference in the resulting equilibrium policies and values. Hence, it's difficult to know when the transition and reward model is accurate enough, since even a small error can mean a completely different equilibrium policy and value.

## 6. Conclusion

This paper examined the general-sum multiagent reinforcement learning algorithm introduced at ICML by Hu and Wellman (1998). The main contribution of their paper is both the algorithm and a proof of its convergence to a Nash equilibrium under specific assumptions. A technique with such guarantees would be very desirable for multiagent reinforcement learning. Unfortunately, the result depends on a crucial lemma, the proof of which we have shown to be incomplete.

We presented both a counterexample to the lemma and described the flaw in their proof. We also presented strengthened assumptions, which would allow the lemma and theorem to be valid, but creates even further restrictions on its applicability. These restrictions were discussed along with other issues surrounding the problem of equilibrium convergence in general-sum stochastic games.

## Acknowledgements

Thanks to Manuela Veloso, Nicolas Meuleau, and Leslie Kaelbling for very helpful discussions and ideas.

## References

Claus, C., & Boutilier, C. (1998). The dyanmics of reinforcement learning in cooperative multiagent systems. *Proceedings of the Fifteenth National Conference on Artificial Intelligence*. Menlo Park, CA: AAAI Press.

Filar, J., & Vrieze, K. (1997). *Competitive Markov decision processes*. New York: Springer Verlag.

Fudenberg, D., & Levine, D. K. (1999). *The theory of learning in games*. Cambridge, MA: The MIT Press.

Hu, J. (1999). *Learning in dynamic noncooperative multiagent systems*. Doctoral dissertation, Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI.

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

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. *Proceedings of the Eleventh International Conference on Machine Learning* (pp. 157–163). San Francisco: Morgan Kaufman.

Nash, Jr., J. F. (1950). Equilibrium points in $n$-person games. *PNAS*, *36*, 48–49. Reprinted in H. W. Kuhn (Ed.). (1997). *Classics in game theory*. Princeton, NJ: Princeton University Press.

Osborne, M. J., & Rubinstein, A. (1994). *A course in game theory*. The MIT Press.

Shapley, L. S. (1953). Stochastic games. *PNAS*, *39*, 1095–1100. Reprinted in H. W. Kuhn (Ed.). (1997). *Classics in game theory*. Princeton, NJ: Princeton University Press.

Watkins, C. J. C. H. (1989). *Learning from delayed rewards*. Doctoral dissertation, King's College, Cambridge, UK.