Learning in Stochastic Games: A Review of the Literature

Serial Number: 1
Department of Computer Science
University of British Columbia
Vancouver, BC, Canada.

Abstract

A great deal of research has been recently focused on stochastic games. Stochastic games can generally model the interactions between multiple agents in an environment. One of the major challenges in stochastic games is the problem of learning a policy to play in such games. While lots of people have worked in this area and proposed various algorithms for that, there is no unique method which can outperform all other methods. Therefore, one should know the pros and cons of different algorithms in the literature to select the one that suits his/her situation best. In this paper, we survey different methods proposed in the literature, group them according to their approaches and discuss about the general advantages and disadvantages of each group.

1 Introduction

The promise of reinforcement learning (RL) is to program an agent in an environment by rewards and punishments and without specifying how the agent should do the task. Even though it dates back to the days of cybernetics, people in machine learning and artificial intelligence communities are really attracted in this area and use it in different situations (Kaelbling et al., 1996). We can see lots of applications of RL in different fields. In optimization, we can see the work by Crites and Barto (1996) who used RL to control a four-elevator, ten-floor system; Or the works by Nie and Haykin (1995) and Zhang and Dietterich (1995) who applied it to dynamic channel allocation and resource constrained scheduling, respectively. In robotics, we can see OBELIX robot (Mahadevan and Connell, 1992) which can push boxes and the autonomous blimp (Rottmann et al., 2007). We can also see other applications of RL in other fields such as game playing, industrial manufacturing, combinatorial search problems, operation research and lots of other areas.

The above applications are all examples of a stationary environment where there is only one agent acting in it. However, in most real world problems, there are multiple agents acting in the environment. These agents can be cooperating with each other, competing against each other, or both. Exploration and mapping (Simmons et al., 2000) is an example of cooperating multi-agent system, in which, multiple agents are trying to find the map of an unknown environment. Setting the price in a marketplace (Tesauro and Kephart, 2002) is an example of a competitive environment. And, a robot soccer team (Balch, 1997) is an example of an environment with both cooperation and competition, because the agents in one team are cooperating while they are competing agents agents in the other team. We can formulate these examples as a stochastic game (Shapley, 1953), in which, agents perform an action and they transition to a new state.

Stochastic games are an important subject of study in both theory and practice. As we can see in the above examples, learning to play in a stochastic game is of high importance. In environments of a stochastic game, we cannot consider the other agents just as part of the environment and use simple RL methods. That's because the

other agents may be learning and adapting their actions based on the actions performed by other agents available in the environment. For example if a company sets its prices regardless of other companies, the other companies may learn how this company is deciding on its prices and set their prices somehow to beat this company. We can see from this example that an agent should explicitly take other agents into account while learning in such environments.

In this paper, we survey the works on learning in stochastic games. We talk about different approaches and algorithms in the literature, group them according to the approach they are using, and generally discuss the advantages and disadvantages of each group.

The rest of the paper is organized as follows: Section 2 provides sufficient background for reader to read the rest of the paper; Section 3 talks about the early works which considered playing in the equilibrium of the game; Section 4 is devoted to reinforcement learning approaches to learn in stochastic games; In section 5, criteria-based methods are introduced; Finally, section 6 concludes the paper and points out some future directions.

2 Background

In this section, we provide the definitions of the terminologies which are used in the rest of the paper so that reader can refer to it when required.

Normal form game (Shoham and Leyton-Brown, 2009): A (finite, n-person) normal form game is a tuple (N, A, u) where N is a set of n players; $A = A_1, A_2, ..., A_n$ where A_i is the actions available to player i; and $u = (u_1, u_2, ..., u_n)$ where u_i is a real-valued utility function for player i.

Stochastic game (Shoham and Leyton-Brown, 2009): A stochastic game is a tuple (Q, N, A, P, R) where Q is a finite set of games; N is finite set of n players; $A = A_1, A_2, ..., A_n$ where A_i is the set of actions available to player i; P is the transition probability function where $P(q, a, \widehat{q})$ is the probability of going from state q to $\widehat{q}$ after profile action a; and $R = r_1, r_2, ..., r_n$ where r_i is a real-valued payoff function for player i.

Stationary strategy (Raghavan and Filar, 1991): A strategy where the player plays in a "memoryless" way, in which, for each matrix game A^s, the player selects a probability distribution on the rows (or columns) of A^s, and every time A^s is reached the rows (or columns) are chosen according to that specific probability distribution.

Pure stationary strategy (Raghavan and Filar, 1991): A strategy where for each matrix game A^s, the player selects a particular row (or column) to be played whenever state s is reached.

Self-play (Shoham and Leyton-Brown, 2009): A situation where all the agents adopt the same strategy (such as reinforcement learning). Note that this does not mean that all the agents always play the same action in each stage game.

3 Playing in the equilibrium

Similar to other forms of games, playing in the equilibrium of the stochastic game is a possible strategy. Shapley (1953) was the first who introduced stochastic games and the equilibria of such games. He considered a zero-sum two-player stochastic game with finite number of states and finite number of pure strategies at each state. While undiscounted stochastic games were proposed later on (Gillette, 1957), Shapley considered a discounted game where the game could end by a positive probability at each step. He proposed the use of stationary strategies for these games by arguing that pure and mixed strategies take into account irrelevant information and it would be better to have a certain behavior strategy in which, no matter what route has been traversed, the same strategy is played at each state. He proved the existence of optimal strategies and the existence of a solution in this settings which means, no matter what the opponent does, player I can gain a certain amount of expected utility by following an optimal stationary strategy.

Takahashi (1963) generalized Shapley's results to the case of infinite states and infinite number of pure strategies at

Fictitious Play Algorithm
Initialize V(s) arbitrarily, $U_i(s, a_i) = 0$ and $C_i(s, a_i) = 0$
Repeat the following steps:
$a = (a_1, a_2)$ where $a_i = argmax_{a_i} \frac{U_i(s,a_i)}{C_i(s,a_i)}$
$C_i(s, a_i) = C_i(s, a_i) + 1$
$U_i(s, a_i) = U_i(s, a_i) + R_i(s, a) + \gamma \left(\sum_{s'} T(s, a, s')V(s')\right)$
$V(s) = max_{a_1} \frac{U_1(s,a_1)}{C_1(s,a_1)}$

Table 1: The algorithm for fictitious play

each state. There were also other works trying to improve Shapley's results. We can consider Shapley's algorithm as an extension of value iteration algorithm (Bertsekas, 1987) for MDPs to stochastic games which iterates through value space. The work by Hoffman and Karp (1966) is an example attempt to improve Shapley's algorithm where they iterate through both value and strategy state. Another well-known work is done by Pollatschek and Avi-Itzhak (1969) where they try to extend the policy iteration algorithm (Howard, 1960) for MDPs to stochastic games. There has been also attempts to calculate the equilibrium for a broader range of games. Among these, we can mention the work by Fink (1964) who proposed a way of calculating the equilibrium for general-sum n-player stochastic games.

Even though various algorithms have been proposed to solve such stochastic games, it has been shown that they lack an algebraic property called "ordered field property" which makes them really difficult to be solved efficiently (Parthasarathy and Raghavan, 1981). Various efforts has been made in the literature to find the classes of stochastic games for which ordered field property holds. Among these classes, we can name stochastic games with perfect information (Gillette, 1957) in which the action space of one of the players in every state is singleton; Single-controller stochastic games (Stern, 1975) in which the new state is just a function of the previous state and the action played by one of the players; And switching-controller stochastic games (Filar, 1981) which is a generalization of single-controller stochastic games in which the new state is a function of the previous state and player 1's action for some states and a function of the previous state and player 2's action for the rest.

3.1 Fictitious Play

Now we talk about about a famous algorithm for finding and playing in the Nash equilibrium of the game called fictitious play (Robinson, 1951; Vrieze, 1987). Fictitious play can find the equilibrium in zero-sum games and some of general-sum games and its extension called smooth fictitious play (Fudenberg and Levine, 1999) can play mixed equilibrium. The algorithm maintains the count and the utility for each action and when it want to choose an action in a given state, it deterministically computes the average expected discounted reward from the past experience and chooses the one that has done best in the past. We can see the algorithm of fictitious play for a two-player zero-sum game in Table 1.

3.2 General Problems

Playing in a Nash equilibrium of the stochastic game may seem to be a good choice of strategy. However, not only its computation is NP-complete for pure stationary strategies and NP-hard for stationary strategies (Ummels and Dominik, 2009), but also it is not a good choice in cases when there are multiple equilibria and we don't know in what equilibrium the opponents are planning to play; as well as when the opponents are not playing in an equilibrium. Consider the case when two players are playing the game of matching pennies infinite times and

Joint-Action Learner Algorithm
Initialize Q(s, a) and V(s) arbitrarily and C(s, a) = 0, n(s) = 0
Repeat the following steps:
Play $argmax_{a_i} \sum_{a_{-i}} \frac{C(s,a_{-i})}{n(s)} Q(s, \langle a_i, a_{-i} \rangle)$
Observe next state s', reward r and opponent's action a_{-i}.
$Q(s,a) = (1-\alpha)Q(s,a) + \alpha(r + \gamma V(s'))$ where $a = \langle a_i, a_{-i} \rangle$ and V(s) is the max possible value in s.
$C(s, a_{-i}) = C(s, a_{-i}) + 1$
$n(s) = n(s) + 1$

Table 2: The algorithm for joint-action learner

suppose the first player is always playing heads. In this case, it is not a good choice for the second player to play in the equilibrium of the game (which randomizes between heads and tails with equal probability). Instead, he has to learn what the other player is doing and then adapt his strategy accordingly to maximizes his payoff.

4 Reinforcement Learning

Markov decision process (MDP) is a useful mathematical framework for decision making under uncertainty. They are actually the single agent version of a stochastic game (Bowling and Veloso, 2000). Reinforcement learning (Sutton and Barto, 1998) is a successful way of finding the optimal policy in MDP framework. There have been also attempts to use reinforcement learning as a basis for learning in stochastic games. Even though there are single agent learning algorithms which can be used for stochastic games (Jaakkola et al, 1994; Baird and Moore, 1999), they usually fail to converge to a policy in stochastic games and we don't talk about them in this paper. Instead, we talk about the works that have directly taken into account the stochastic games and proposed reinforcement learning algorithms to learn in these games.

4.1 Joint-Action Learners

Early attempts were trying to model the opponent. They assumed the opponent is playing according to a stationary distribution over his actions and learned an explicit model of it. At the same time, using temporal difference (Sutton and Barto, 1990), joint-action values were learned and both these were used to select an action to play. This has been done in a fully collaborative domain by Claus and Boutilier (1998) where they call it joint-action learner (JAL) and in a fully competitive domain by Uther and Veloso (1997). The general algorithm for it can be seen in Table 2 (C(s, a) counts the number of times that action "a" has been played in state "s" by the opponent and n(s) counts the total number of times that state "s" has been met). While the policy learned by this algorithm is optimal when other agents are following a stationary policy, the major problem with this algorithm is that it cannot converge to mixed equilibria even in self-play which means that it is not convergent.

4.2 minmaxQ and NashQ

Another reinforcement learning approach to learn in stochastic games is minmaxQ (Littman, 1994). MinmaxQ is an extension of the traditional Q-Learning to zero-sum stochastic games. It extends the Q-values to maintain the joint action, uses linear programming to update V-values and find the policy, and it uses a decreasing learning rate. The algorithm for minmaxQ is in Table 3. While minmaxQ was proposed for zero-sum stochastic games, Hu and Wellman (1998) extended this algorithm to general-sum stochastic games. They called their algorithm NashQ in

MinmaxQ Algorithm
Initialize $Q(s,a)=1, V(s)=1, \Pi(s,a_i)=\frac{1}{\lvert A_i\rvert}, \alpha=1$
Repeat the following steps:
Play action a_i in state s
Observe new state s' reward r and opponent's action a_{-i}
$Q(s,a)=(1-\alpha)*Q(s,a)+\alpha(r+\gamma V(s'))$ where $a=\langle a_i, a_{-i}\rangle$
Use linear programming to find $\Pi(s,a_i)=argmax_{\Pi'(s,a_i)} min_{a_{-i}} \sum_{a'_i} \Pi(s,a'_i)*Q(s,a'_i,a_{-i})$
$V(s)=min_{a_{-i}} \sum_{a'_i} \Pi(s,a'_i)*Q(s,a'_i,a_{-i})$
Decay α

Table 3: The algorithm for minmaxQ

which, they assume each player can observe the immediate rewards and previous action of the other player. Then each agent, not only keeps his own Q values (like minmaxQ) but also maintains the opponent's Q-values. While minmaxQ uses a linear programming to find the solution, NashQ uses quadratic programming.

This algorithm is the first algorithm based on reinforcement learning which is addressing the problem of learning in general-sum stochastic games. Like other rudimentary algorithms for different problems, they make some restrictive assumptions such as limiting the structure of intermediate matrix games, restricting the game to have just one equilibrium, considering that every state and action have been infinitely visited, etc. which decreases its applicability to various problems.

These two algorithms have the nice property of convergence. However, their problem is that they will not choose the optimal policy when the opponent is not playing in the equilibrium of the game.

5 Criteria based methods

One of the problems with learning algorithms in stochastic games is that there are not predefined criteria so that people try to address them in the algorithms they propose. We can see some works in the literature which they defined some criteria of their own and they tried to propose an algorithm which works best according to those criteria. In this section. we will see some of these works and will discuss about the criteria they considered and the algorithms they proposed.

5.1 Rationality and Convergence

Bowling and Veloso (2001) were among the first who defined some criteria for effective learning. They defined rationality and convergence as follows:

"Rationality: If the other players' policies converge to stationary policies then the learning algorithm will converge to a policy that is a best-response to their policies"

"Convergence: The learner will necessarily converge to a stationary policy"

They considered the above definition of convergence in the self-play and claimed that none of the algorithms proposed up to that time satisfied both criteria. We can see this claim in the algorithms we discussed in previous sections. For example, we can see that minmaxQ is convergent but not rational and JAL is rational but not convergent. They proposed WoLF (Win or Learn Fast) policy hill climbing algorithm in their paper which could address both these problems.

First of all, they introduce policy hill climbing which is an extension of Q-learning to play mixed strategies. This algorithm, maintains Q values and current policy. Then it improves the policy by performing hill climbing in the

space of mixed strategies. Like Q-learning this is a rational but not convergent learning method. Then, they propose WoLF policy hill climbing based on this algorithm. The idea behind the new algorithm is to use a variable learning rate which can have a high or a low value. The high value is used when the player is losing and low value is used when the player is winning.
Since the only difference between this algorithm and the simple policy hill climbing is the learning rate, this algorithm is also rational. The authors also prove in another paper (Bowling and Veloso, 2001) that using WoLF, the algorithm is guaranteed to converge to a Nash equilibrium is self-play. This means that their algorithm satisfies both criteria they proposed. They also represented this result empirically for some of the benchmark games such as soccer and grid-world.
One of the major problems with WoLF is that it is only applicable to two-player games. Even though this problem was addressed by Conitzer and Sandholm (2003) later on, there are some problems with this algorithm. The first problem is that it only considers the convergence in self-play, while in reality there may be both self-play and stationary opponents. The second problem is that they require to converge to a Nash equilibrium while in some games like the Prisoner's Dilemma, players may not be willing to play in the equilibrium of the game but they would rather cooperating as much as the other player is not defecting.

5.2 Scalability

Bowling and Veloso (2002) realized that lots of the works in the literature have been applied only to small games with at most hundreds of states. Even though there were some algorithms proposed for some specific large games such as Checkers (Samuel, 1967) and TD-Gammon (Tesauro, 1995), they could only be applied to these games and they could only play a deterministic policy. Therefore, Bowling and Veloso (2002) decided to scale some of the methods based on Nash equilibrium to games with intractable state spaces. Based on their arguments in their paper, we can define the scalability criteria as the ability of an algorithm to be applied to intractable state spaces.
They used the idea of generalization and approximation and combined three different methods to propose one algorithm. First of all, they used tile coding (Sutton and Barto, 1998) which given a set of continuous features, creates a set of boolean features. In order to keep the number of parameters manageable, they hashed each of the tiles to a fixed size table. Then they used a policy gradient technique (Sutton et al., 2000) with a variable learning rate based on WoLF (Bowling and Veloso, 2001) to learn a policy for each of the states in the hash table. Policy gradient technique makes the algorithm not to be deterministic and using WoLF makes it convergent. They applied their method to the two-player n-card version of Goosfiel game invented by Flood (1985) which has more that 10^{11} states and needs approximately 2.5TB of storage to store Q values and the policy if we use simple Q-learning.
Even though they claim based on their results that this method can be applied to large games, using hash tables puts doubt on its efficiency. They use a hash table of size 10^6 which means that, on average, 10^5 states are mapped to one cell of the hash table. We know that hash functions do not necessarily map similar states to a particular cell in the table and this means that their algorithm is learning one policy for 10^5 potentially different states.

5.3 Other Criteria

There have been lots of other works in the literature which have defined some new criteria and proposed algorithms to address them. Fudenberg and Levine (1995) introduced the following criteria:
"Safety: The learning rule must guarantee at least the minimax payoff of the game."
"Consistency: The learning rule must guarantee that it does at least as well as the best response (in the stage game) to the empirical distribution of play when playing against an opponent whose play is governed by independent draws from any fixed distribution."
Since neither these criteria nor the criteria considered by Bowling and Veloso (2001) or others in the literature con-

sidered future play of the opponent, new criteria called targeted optimality, auto-compatibility and a new definition of safety were proposed by Powers and Shoham (2005) to address this problem. Furthermore, In order to develope algorithms that can strongly guarantee payoffs against a variety of opponents and also cooperate with one another, two new criteria called targeted group optimality and a new definition of safety were proposed by Vu et al. (2006). Even though these are really interesting criteria and there have been interesting algorithms proposed for them, to keep things simple, we don't talk about these criteria and algorithms in this paper.

6 Conclusion

Stochastic games have been an interesting subject of study from the time they have been introduced until now. Lots of people have tried to propose algorithms to learn how to play in such games. There have been both game theory and reinforcement learning approaches to learn to play in this games. In this paper we examined some of the works in the literature and discussed generally about their advantages and disadvantages.

Recent works are trying to propose some criteria for effective learning in stochastic games and then try to propose an algorithm which addresses these criteria. However, we can see that people often argue that there are problems with the criteria considered by other papers when they want to propose new ones. In future, it would be better if people, first of all, try to just propose suitable criteria without proposing any algorithm for that. Once all people agreed on a common set of criteria, then thay can try to propose algorithms that address all of at least of those criteria.

One of the other major challenges in learning in stochastic games is how to evaluate the model. While there are some benchmarks for small games, they are not useful if we want to compare the efficiency of two algorithms. People often have two learning algorithms learn against each other and examine the reward over time. However, this cannot be used for methods which learn in self-play. Furthermore, there is no benchmark and a unique way of evaluation for large games. For example, we can see that Bowling and Veloso (2002) use a non-benchmark game to test their proposed algorithm and train a challenger against their agent to find the worst-case performance of their policy. However, having a good worst-case does not necessarily mean that the algorithm can play a game optimally. A future direction would be trying to come up with a unique and efficient way of testing and evaluating such algorithms so that one can easily compare his method to others.

References

Baird, L.C. and Moore, A.W. 1999. *Gradient descent for general reinforcement learning*. In Advances in Neural Information Processing Systems11, The MIT Press.

Balch, T. 1997. *Learning roles: Behavioral diversity in robot teams*. AAAI Workshop on Multiagent Learning.

Bertsekas, D.P. 1987. *Dynamic Programming: Deterministic and Stochastic Models*. Prentice-Hall.

Bowling, M. and Veloso, M. 2000. *An analysis of stochastic game theory for multiagent reinforcement learning*. No. CMU-CS-00-165. CARNEGIE-MELLON UNIV, PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE.

Bowling M. and Veloso, M. 2001. *Rational and convergent learning in stochastic games*. International joint conference on artificial intelligence 17(1), pp. 1021-1026.

Bowling M. and Veloso, M. 2001. *Convergence of gradient dynamics with a variable learning rate*. In Proceedings of the Eighteenth International Conference on Machine Learning.

Bowling M. and Veloso, M. 2002. *Scalable learning in stochastic games*. AAAI Workshop on Game Theoretic and Decision Theoretic Agents, pp. 11-18.

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

Conitzer, V. and Sandholm, T. 2003. *The dynamics of reinforcement learning in cooperative multiagent systems*. In Proceedings of the Fifteenth National Conference on Artificial Intelligence, pp. 746-752.

Crites, R.H. and Barto, A.G. 1996. *Improving Elevator Performance Using Reinforcement Learning*. Advances in Neural Information Processing Systems, Proceedings of the 1995 Conference, pp. 1017-1023.

Filar, J.A. 1981. *Ordered Field Property for Stochastic Games When the Player Who Controls Transitions Changes from State to State*. Journal of Optimization Theory and Applications 34(4), pp. 503-515.

Fink, A.M. 1964. *Equilibrium in a stochastic n-person game*. Hiroshima Mathematical Journal 28(1), pp. 89-93.

Flood, M. 1985. *Interview by Albert Tucker*. The Princeton Mathematics Community in the 1930s, Transcript Number 11.

Fudenberg, D. and Levine, D. 1995. *Universal consistency and cautious fictitious play*. Journal of Economic Dynamics and Control 19, pp. 1065-1089.

Fudenberg, D. and Levine, D.K. 1999. *The Theory of Learning in Games*. The MIT Press.

Gillette, D. 1957. *Stochastic Games with Zero Stop Probabilities*. Contributions to the Theory of Games 3, pp. 179-187.

Hoffman, A.J. and Karp, R.M. 1966. *On Non-terminating Stochastic Games*. Journal of Management Science 12, pp. 359-370.

Howard, R.A. 1960. *Dynamic Programming and Markov Processes*. The MIT Press.

Hu, J. and Wellman, M.P. 1998. *Multiagent reinforcement learning: Theoretical framework and an algorithm*. Proceedings of the fifteenth international conference on machine learning 242.

Jaakkola, T., Singh, S.P., and Jordan, M.I. 1994. *Reinforcement learning algorithm for partially observable Markov decision process*. In Advances in Neural Information Processing Systems 6.

Kaelbling, L.P., Littman, M.L., and Moore, A.W. 1996. *Reinforcement Learning: A Survey*. arXiv preprint cs/9605103.

Littman, M.L. 1994. *Markov games as a framework for multi-agent reinforcement learning*. In Proceedings of the Eleventh International Conference on Machine Learning 157, pp. 157-163.

Mahadevan, S. and Connell, J. 1992. *Automatic programming of behavior-based robots using reinforcement learning*. Artificial Intelligence 55(2-3), pp. 311-365.

Nie, J. and Haykin, S. 1995. *A Dynamic Channel Assignment Policy Through Q-learning*. CRL Report 334, Intelligence (IJCAI-95), Morgan Kaufmann, pp. 1114-1120.

Parthasarathy, T. and Raghavan, T.E.S. 1981. *An Orderfield Property for Stochastic Games when One Player Controls Transition Probabilities*. Journal of Optimization Theory and Applications 33(3), pp. 375-392.

Pollatschek, M. and Avi-Itzhak, B. 1969. *Algorithms for Stochastic Games with Geometrical Interpretation*. Journal of Management Science (15), pp. 399-415.

Powers, R. and Shoham, Y. 2005. *New criteria and a new algorithm for learning in multi-agent systems*. In Advancesin Neural Information Processing Systems 17.

Raghavan,T.E.S. and Filar, J.A. 1991. *Algorithms for stochastic games - a survey*. Zeitschrift für Operations Research 35(6), pp. 437-472.

Robinson, J. 1951. *An iterative method of solving a game*. Annals of Mathematics 54, pp. 296-301.

Rottmann, A., Plagemann, C., Hilgers, P., and Burgard, W. 2007. *Autonomous blimp control using model-free reinforcement learning in a continuous state and action space*. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS.

Samuel, A.L. 1967. *Some studies in machine learning using the game of checkers*. IBM Journal on Research and Development 11, pp. 601-617.

Shapley, L.S. 1953. *Stochastic games*. proceedings of national academy of sciences 39, pp. 1095-1100.

Shoham, Y. and Leyton-Brown, K. 2009. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press.

Simmons, R., Apfelbaum, D., Burgard, W., Fox, D., Moors, M., Thrun, S., and Younes, H. 2000. *Coordination for multi-robot exploration and mapping*. In Proceedings of the National conference on Artificial Intelligence, pp. 852-858.

Stern, M. 1975. *On Stochastic Games with Limiting Average Payoff*. PhD Thesis, University of Illinois at Chicago.

Sutton, R.S. and Barto, A.G. 1990. *Time-derivative models of Pavlovian reinforcement*. In Learning and computational neuroscience: Foundations of adaptive networks.

Sutton, R.S. and Barto, A.G. 1998. *Reinforcement Learning*. The MIT Press.

Sutton, R.S., McAllester, D., Singh, S., and Mansour, Y. 2000. *Policy gradient methods for reinforcement learning with function approximation*. Advances in Neural Information Processing Systems 12, MIT Press.

Takahashi, M. 1963. *Stochastic Games with Infinitely Many Strategies*. Hiroshima Mathematical Journal 26(2), pp. 123-134.

Tesauro, G.J. 1995. *Temporal difference learning and TD-Gammon*. Communications of the ACM 38, pp. 48-68.

Tesauro, G. and Kephart, L.O. 2002. *Pricing in agent economies using multi-agent Q-learning*. Game Theory and Decision Theory in Agent-Based Systems, Springer, pp. 293-313.

Ummels, M. and Dominik, W. 2009. *The Complexity of Nash Equilibria in Simple Stochastic Multiplayer Games*. Automata, Languages and Programming. Springer Berlin Heidelberg, pp. 297-308.

Uther, W. and Veloso, M. 1997. *Adversarial reinforcement learning*. Technical report, Carnegie Mellon University.

Verieze, O.J. 1987. *Stochastic Games with Finite State and Action Spaces*. CWI tracts 33, pp. 1-221.

Vu, T., Powers, R., and Shoham, Y. 2006. *Learning against multiple opponents*. Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems, ACM.

Zhang, T.G. and Dietterich, W. 1995. *A Reinforcement Learning Approach To Job-Shop Scheduling*. In Proceedings of the 14th International Joint Conference on Artificial Intelligence pp. 1114-1120.