LEARNING FROM DELAYED REWARDS USING INFLUENCE VALUES APPLIED TO COORDINATION IN MULTI-AGENT SYSTEMS

Dennis Barrios-Aranibar*, Luiz Marcos Garcia Gonçalves*

*Department of Computing Engineering and Automation
Federal University of Rio Grande do Norte
Lagoa Nova 59.072-970 - Natal - RN - Brazil

Emails: dennis@dca.ufrn.br, lmarcos@dca.ufrn.br

Abstract— In this work we propose a new paradigm for learning coordination in multi-agent systems. This approach is based on social interaction of people, specially in the fact that people communicate to each other what they think about their actions and this opinion has some influence in the behavior of each other. We propose a model in which multi-agents learn to coordinate their actions giving opinions about the actions of other agents and also being influenced with opinions of other agents about their actions. We use the proposed paradigm to develop a modified version of the Q-learning algorithm. The new algorithm is tested and compared with independent learning (IL) and joint action learning (JAL) in a grid problem with two agents learning to coordinate. Our approach shows to have more probability to converge to an optimal equilibrium than IL and JAL Q-learning algorithms, specially when exploration increases. Also, a nice property of our algorithm is that it does not need to make an entire model of all joint actions like JAL algorithms.

Keywords— Influence Value, Reinforcement Learning, Multi-agent coordination.

1 Introduction

As Kok and Vlassis postulate, a multi-agent system (MAS) consists of a group of agents that can potentially interact with each other (Kok and Vlassis, 2004). However, this interaction has to be coordinated. Coordination, collaboration and cooperation are three terms used without distinction when working with multi-agent systems. In this paper, we adopt a definition proposed by Noreils (Noreils, 1993) in which cooperation occurs when several agents (or robots) are gathered together so as to perform a global task. Coordination and collaboration are two forms of cooperation (Botelho and Alami, 2000). Coordination occurs when an entity coordinates its activity with another, or it synchronizes its action with respect to the other entity, by exchanging information, signals, etc. And, collaboration occurs when agents decompose the global task in subtasks and each subtask is performed by a specific agent.

The focus of this work is coordination, thus, the problem is how to make agents perform their actions according to the other agents actions in order to achieve a Nash equilibrium (Kononen, 2004).

If N is the number of players, the strategies $\sigma_1^*, \ldots, \sigma_N^*$ constitute a *Nash equilibrium* solution of the game if the following inequality holds for all $\sigma_i \in \Sigma_i$ and for all i:

$$r_i(\sigma_1^*, \ldots, \sigma_{i-1}^*, \sigma_i, \sigma_{i+1}^*, \ldots, \sigma_N^*) \leq r_i(\sigma_1^*, \ldots, \sigma_N^*) \quad (1)$$

The idea of the Nash equilibrium solution is that the strategy chosen by each player is the best response to his opponents' play and therefore there is no need for deviation from this equilibrium point for any player alone (Kononen, 2004).

Reinforcement learning has been widely applied to the coordination problem. The approaches proposed until now can be classified into two paradigms: independent learning and joint action learning. Independent learning agents learn values of individual actions considering only a global reward received by the agent. On the other hand, joint actions learning agents learn values of joint actions considering a global reward and a model of the other agents behavior.

Our approach is based in the idea that an agent learns only the values of their own actions, but it considers the global reward and the influence that other agents have over the learning agent. This approach does not fit into none of the both classifications explained previously, thus, we are proposing a new paradigm that we call "Influence Valued Reinforcement Learning".

The two paradigms for coordination in multi-agent systems are better explained in section 2. Then, our proposed approach is introduced in section 3 and its experimental results in comparison with the other approaches are shown in section 4. Finally, its contribution and applications are discussed in section 5.

2 Coordination in Multi-Agent Systems

The problem of coordination in multi-agent systems has had increasingly attention by the artificial intelligence community. It is not difficult to find solutions based on reinforcement learning (Kok and Vlassis, 2004; Kononen, 2004).

Traditional solutions based on reinforcement learning can be classified into independent learning and joint action learning. Both paradigms intend to achieve the Nash equilibrium. However, in certain games there exist several equilibrium points and this became a challenge for reinforce-

ment learning researches. This occurs because when a problem has several equilibrium points the concept of optimal equilibrium appears. When all agents in the system are evolutionary, it is difficult to achieve an optimal equilibrium because of the uncertainty of actions of other agents.

2.1 Independent Learning

The basic idea of independent learning (IL) is that agents learn independently as if other agents does not exist. Thus, an agent only matters with the reward obtained from the environment and not with the actions that may be performed by other agents nor with the relation between its actions and other agents actions. In this sense, traditional reinforcement learning algorithms can be applied without any modification. Claus and Boutilier (Claus and Boutilier, 1998) shows empirically that this kind of solution converges to a Nash equilibrium, but, depending on the structure of the problem to be solved, it can not converge to the optimal Nash equilibrium. Because independent learning does not consider information about other agents, in certain problems it could be inefficient when trying to converge to an optimal equilibrium.

In this work, we are interested in multi state problems with two agents, thus, we select to use learning from delayed rewards algorithms. In the case of IL we use the Q-Learning (IQ-Learning) algorithm defined by the following equation:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) +$$

$$\alpha(r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)) \qquad (2)$$

where $Q(s_t, a_t)$ is the value of the action a_t in the state s_t, α is the learning rate ($0 \leq \alpha \leq 1$), γ is the discount rate ($0 \leq \gamma \leq 1$), s_{t+1} is the resulting state of executing the action a_t and r is the instantaneous reward obtained by executing the action a_t.

When using independent learning in multi-agent systems, one of the main issues is the state representation. That is, to decide if the state in the reinforcement learning algorithm represents the individual state of the agent or the global state of all agents in the system. In this work we explore both possibilities although the most used one is the representation of the global state of the system.

2.2 Joint Action Learning

The basic idea of joint action learning (JAL) is that agents do not calculate the value of their actions. Instead of that, they calculate the values of their actions when combined with actions of other agents. Each combination is known as a joint action. Also, the agent decide the action to perform based on the actions that other agents will probably execute. In this sense, each agent has to construct a model of the behavior of the other agents.

Because joint action learners does not have an entire model of joint actions, it becomes difficult to implement this approach when number of agents, states and/or actions increase.

For the joint actions learning paradigm, we use an algorithm based on Q-Learning (JAQ-Learning).

In the JAQ-Learning algorithm the value of a joint action (a, b) of an agent i is modified by the equation 3.

$$Q_i(s_t, a_t, b_t) \leftarrow Q_i(s_t, a_t, b_t) +$$

$$\alpha(r_{t+1} + \gamma \max_{a,b} Q_i(s_{t+1}, a, b) - Q_i(s_t, a_t, b_t)) \quad (3)$$

where a_t is the action performed by the agent i at time t, b_t is the action performed by the other agent, $Q_i(s_t, a_t, b_t)$ is the value of the joint action (a_t, b_t) for agent i in the state s_t, r_{t+1} is the reward obtained by agent i as it executes action a_t and as the other agent executes action b_t, α is the learning rate ($0 \leq \alpha \leq 1$) and γ is the discount rate ($0 \leq \gamma \leq 1$)

However, an agent has to decide between its actions and not between joint actions. For this decision, it uses the expected value of its actions that include information about the joint actions and current beliefs about other agent (Equation 4).

$$EV(s_t, a_t) \leftarrow \sum_{b \in B} Q(b \cup a_t) * Pr_t(b) \qquad (4)$$

where a_t is the action performed by the agent, $EV(s_t, a_t)$ is the expected value of action a_t in state s_t, b is an action of the other agent, B is the set of actions of the other agent and $Pr_t(b)$ is the probability that other agent performs action b in state s_t.

2.3 Exploration Strategy

An important decision to take when working with any kind of reinforcement learning algorithm is the action selection strategy. The selected strategy has to guarantee both exploration and exploitation. In this sense, the best one is the *softmax* strategy.

A popular softmax strategy is the one based on Boltzman equation:

$$Pr(a) = \frac{e^{Q(a)/T}}{\sum_{a'} e^{Q(a')/T}} \qquad (5)$$

where T is the temperature parameter that can be decreased over time so that the exploitation probability increases (and can be done in a such way that convergence is assured)(Singh et al., 2000).

3 Influence Valued Reinforcement Learning

As said, our paradigm is based on social interaction of people. When two persons interact, they communicate to each other what they think about their actions. Thus, if a person A does not like an action performed by another person B, then A may protest, gently, against B. If the person B continues doing the same action, then A gets angry and angrily protest against B. Note that the protesting force is proportional to the number of times the action is repeated. At some time, person A may eventually fight against B.

On the other hand, if a person A likes the action performed by another person B, then A praises B. Also if the performed action is very good, then person A praises B a lot. Note that if B continues to perform this action, then A will be accustomed and with time A will stops praising B. This means that the praising force is inversely proportional to the number of times the action is repeated.

We also note that protests and praises of other people can *influence* the behavior of a person. When other people protests against us, we try to avoid actions that caused these protests and when the opposite occurs (people praises us), we try to repeat the same actions.

Inspired in the fact explained above, we propose a new paradigm for machine learning denominated "Influence Valued Reinforcement Learning". In our approach, agents calculate the value of their individual actions based on a global reward (reward given by the environment) and on a value called influence value.

The influence value for an agent is calculated by the product of an influence rate ($0 \leq \beta \leq 1$) and the *opinion* of other agents have about agent's action.

The influence rate (β) tells if the agent is or not influenced by the opinion of other agents. *Opinion* is the value that other agents have about the action of an agent. If the instantaneous reward that the agent receives at a certain time plus the value of the new state that the agent reaches is greater than the value of its own action, the *opinion* about the actions performed by the other agents is positive and inversely proportional to the times that the other agents performed the actions. If the reward that the agent receives plus the value of the new state is lesser than the value of its own action, the *opinion* about the actions performed by other agents is negative and directly proportional to the times that the other agents performed the actions.

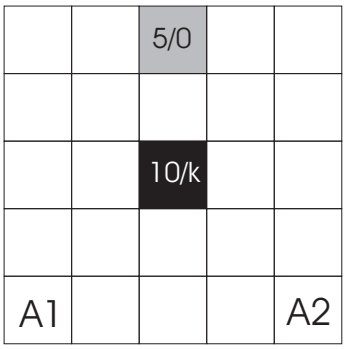

Figure 1: *Grid World* game for testing coordination between two agents.

3.1 Influence Value Q-Learning

The IVQ-learning algorithm is a Q-Learning based algorithm developed using the Influence Valued Reinforcement Learning paradigm. In this sense the action value for the delayed reward in the two agents problem is modified using the Equation 6.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) + \beta(Op_B)) \quad (6)$$

where $Q(s_t, a_t)$ is the value of action a_t done by agent A, α is the learning rate ($0 \leq \alpha \leq 1$), γ is the discount rate ($0 \leq \gamma \leq 1$), β is the influence rate ($0 \leq \beta \leq 1$), thus, β is the influence of the opinion of the other agent (B) over the value of action a_t and Op_B is the opinion that the other agent has about action a_t in the state s_t.

The opinion of an agent about an action performed by another agent is calculated by:

$$Op_B = \begin{cases} (r_b + \max_b Q(s_{t+1}, b) - Q(s_t, b_t)) * Pe(s_t, a_t) \\ \text{if } (r_b + \max_b Q(s_{t+1}, b) - Q(s_t, b_t)) < 0 \\ (r_b + \max_b Q(s_{t+1}, b) - Q(s_t, b_t))/Pe(s_t, a_t) \\ \text{if } (r_b + \max_b Q(s_{t+1}, b) - Q(s_t, b_t)) > 0 \\ 0 \\ \text{in other case} \end{cases}$$

where r_b is the instantaneous reward obtained by the agent B, $Q(s_t, b_t)$ is the value of the action b_t of agent B in the state s_t and $Pe(s_t, a_t)$ is the percentage of times that the agent A performs action a_t in state s_t.

We note that independent learners have to store only the values of their individual actions; joint action learners have to store the values of joint actions and the probability that other agents execute their actions. The influence valued learners have to store the values of their individual actions and the percentage that other agents execute their actions. Thus, the number of stored values increase as the number of actions and/or the number of agents increase. In this sense, independent learning and influence valued learning have an advantage over joint action learning.

4 Experimental Results

In order to test our approach in comparison with the existing ones, we create a game we called the *grid world* game where the goal is coordination between two agents. That is, in this game (figure 1) both agents have to coordinate their actions in order to obtain positive rewards. Lack of coordination causes penalties for both.

The game starts with the agent one ($A1$) in position $(5, 1)$ and agent two ($A2$) in position $(5, 5)$. The idea is for them to reach positions $(1, 3)$ and $(3, 3)$ in order to finish the game. If they reach these final positions at the same time, they obtain a positive reward. When they reach the position $(1, 3)$ at the same time they obtain 5 points and when they reach the position $(3, 3)$ at the same time, they obtain 10 points. However, if only one of them reaches the position $(3, 3)$ they are punished with a penalty value k. In the other case, if only one of them reaches position $(1, 3)$ they are not punished.

The game ends when at least one of the agents reaches the positions $(1, 3)$ and $(3, 3)$ or, also, the game ends if the action of at least one of them leaves it out of the grid. The possible actions are four in the game: go right, go left, go up and go down. For example, if the position of an agent is at the cel $(2, 2)$ then the action go right brings the agent to the position $(2, 3)$. The action go left brings it to the position $(2, 1)$, go up brings it to position $(1, 2)$, and finally the action go down leads the agent to position $(3, 2)$.

This game has several Nash equilibrium solutions, the policies that lead agents to obtain 5 points and 10 points, however, optimal Nash equilibrium solutions are those that lead agents to obtain 10 points in four steps.

We use the same game to test four implemented versions of the Q-Learning algorithm. The first two implemented algorithms are versions of the independent learning paradigm, the third one is a version of the joint actions learning paradigm and the last one is our paradigm, the influence valued learning.

The first algorithm (Independent Learning A) considers that each agent only learns the values of its individual actions without considering the actions performed by the other agents. The state in this version of the algorithm is the position of the agent, thus the state space does not consider the position of the other agents. The second version of this algorithm (Independent Learning B) also considers the individual actions, but, the state space in this version of the algorithm is the position of both agents. The third one (Joint Actions Learning) also considers that the state is formed by the positions of both agents and learns the joint actions for each state. The last one (Influence Valued Learning), that we propose, also considers the same kind of state plus the influence values given by other agents.

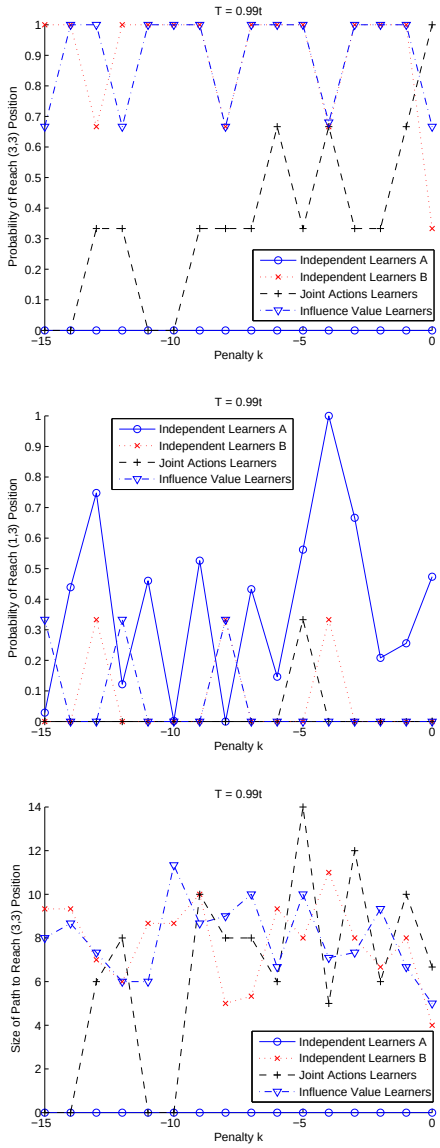

Figure 2: Comparison of Algorithms Performance with $\alpha = 1$, $\lambda = 0.1$, $\beta = 0.1$ and $T = 0.99t$

In the tests, each learning algorithm is executed three times for each value of penalty k $(0 \leq k \leq 15)$ and using five different decreasing rates of temperature T for the softmax policy $(0.99t, 0.995t, 0.999t, 0.9995t, 0.9999t)$. Each resulting policy (960 policies, 3 for each algorithm with penalty k and a certain decreasing rate of T) was tested 1000 of times.

Different penalties k were chosen for testing the capability of each algorithm to reach the optimal equilibrium (in our case the position $(3, 3)$, at the same time, and in four steps). Different values for the decreasing rate of temperature T was chosen in order to test the influence of exploration $\times$ exploitation over each algorithm while trying to reach the optimal equilibrium.

The statistical mean of the percentage of

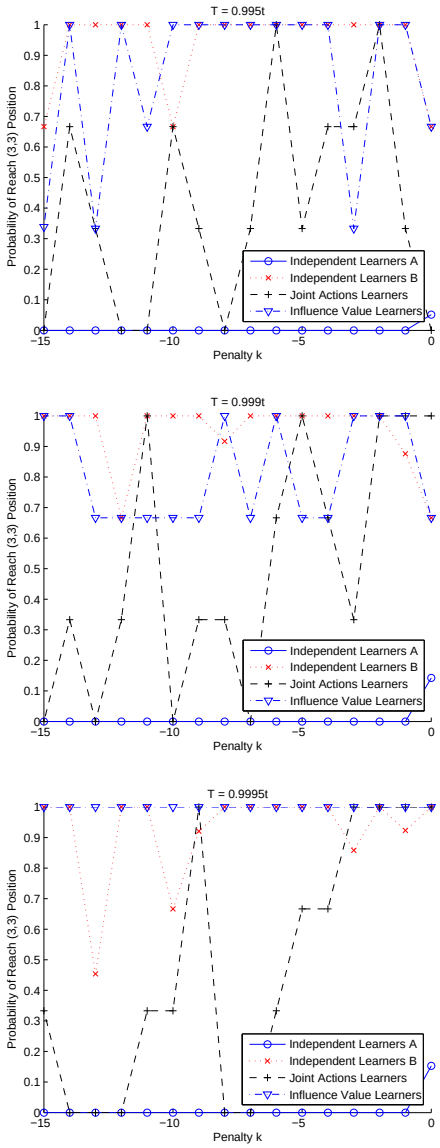

Figure 3: Probability of Reaching $(3,3)$ Position at the Same Time with $\alpha = 1$, $\lambda = 0.1$, $\beta = 0.1$

times that the position $(3,3)$ is reached at the same time by using each kind of algorithm with a certain penalty k and temperature decreasing rate is obtained. Figure 2 shows the probability of reaching the position $(3,3)$ with $\alpha = 1$, $\lambda = 0.1$, $\beta = 0.1$ and $T = 0.99t$ using the four algorithms. In this figure, we can see that the independent learning algorithm that considers the state as being only the individual positioning of the agent does not have conditions to reach the position $(3,3)$ but, as showed in figure 2, it can reach the position $(1,3)$. This occurs because $(1,3)$ position does not have penalties.

Also it was observed that in this problem the joint action learning algorithm has the smaller probability of convergence to the $(3,3)$ position. This behavior is repeated for the other temperature decreasing rates (figures 3 to 4).

From the experiments, we note that the Independent Learning B and our approach have had almost the same behavior. But, when the exploration rate increases, the probability of convergence to the optimal equilibrium decreases for the Independent Learners and increase for our paradigm. Also, it could be observed that at some times the Joint Actions Learners algorithm can not converge to any equilibrium and that at another times it converges to a policy in which agents can not finish the game.

As shown in figure 4, when exploration rate increases the Independent Learning Algorithm looses the capability of convergence to positions $(1,3)$ and $(3,3)$.

As shown in figures 2 and 4 as more exploratory the action selection policy is, smaller is the size of the path for reaching $(3,3)$ position. When exploration increases, the probability of the algorithms to reach the optimal equilibrium increases too. It is important to note that our paradigm has the best probability of convergence to the optimal equilibrium. One can conclude that by joining the probability of convergence to the position $(3,3)$ and the mean size of the path for reaching this position.

5 Conclusions

In this paper, we propose a new paradigm for learning coordination in multi-agent systems inspired on the simple fact that people in a society interact to each other by exchanging their opinion about their acts. The proposed IVQ-Learning algorithm, developed using this approach has shown to be better than the algorithms based on the models Independent Learning and Joint Action Learning.

Experiments show that our approach has the best probability of convergence to the optimal equilibrium (to reach $(3,3)$ position in four steps). It can reach the optimal equilibrium with the best probability when exploration increases ($T = 0.9999t$). This can be concluded by observing that, at this point, the algorithms find paths of size near the optimal (4), but our paradigm is the one that has the best probability to reach $(3,3)$ position.

We remark that the results obtained are based on the simplest implementation forms of the paradigms. As there exist ways to improve the behavior of the other paradigms (Kapetanakis and Kudenko, 2002; Suematsu and Hayashi, 2002; Tumer et al., 2002; Chalkiadakis and Boutilier, 2003; Sen et al., 2003; Kononen, 2004), it is also possible to find similar methodologies to improve the performance of our new paradigm, thus being it still a better choice.

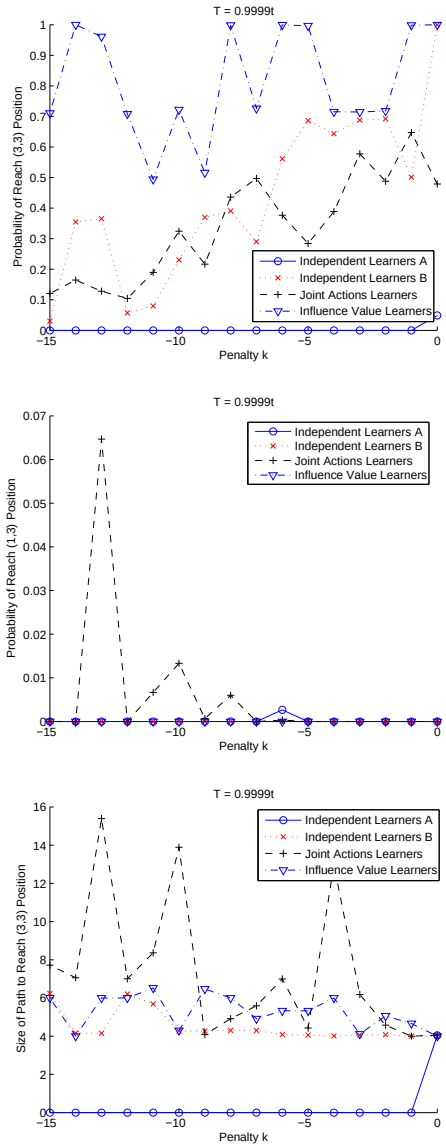

Figure 4: Comparison of Algorithms Performance with $\alpha = 1$, $\lambda = 0.1$, $\beta = 0.1$ and $T = 0.9999t$

Thanks

This work is supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico CNPq/Brasil.

References

Botelho, S. and Alami, R. (2000). Robots that cooperatively enhance their plans, *Proc. of 5th International Symposium on Distributed Autonomous Robotic Systems (DARS 2000). Lecture Notes in Computer Science*, Springer Verlag.

Chalkiadakis, G. and Boutilier, C. (2003). Coordination in multiagent reinforcement learning: a bayesian approach, *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, Melbourne, Australia.

Claus, C. and Boutilier, C. (1998). The dynamics of reinforcement learning in cooperative multiagent systems, *Proceedings of the 15th National Conference on Artificial Intelligence -AAAI-98*, AAAI Press., Menlo Park, CA, pp. 746 – 752.

Kapetanakis, S. and Kudenko, D. (2002). Reinforcement learning of coordination in cooperative multi-agent systems, *Eighteenth national conference on Artificial intelligence*, Edmonton, Alberta, Canada, pp. 326 – 331.

Kok, J. R. and Vlassis, N. (2004). Sparse cooperative q-learning, *Proceedings of the twenty-first international conference on Machine Learning*, Banff, Alberta, Canada, p. 61.

Kononen, V. (2004). Asymmetric multiagent reinforcement learning, *Web Intelligence and Agent System* **2**(2): 105 – 121.

Noreils, F. R. (1993). Toward a robot architecture integrating cooperation between mobile robots: Application to indoor environment, *The International Journal of Robotics Research* **12**(2): 79 – 98.

Sen, S., Airiau, S. and Mukherjee, R. (2003). Towards a pareto-optimal solution in general-sum games, *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, Melbourne, Australia.

Singh, S. P., Jaakkola, T., Littman, M. L. and Szepesvári, C. (2000). Convergence results for single-step on-policy reinforcement-learning algorithms, *Machine Learning* **38**(3): 287–308.

Suematsu, N. and Hayashi, A. (2002). A multiagent reinforcement learning algorithm using extended optimal response, *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, Bologna, Italy.

Tumer, K., Agogino, A. K. and Wolpert, D. H. (2002). Learning sequences of actions in collectives of autonomous agents, *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, Bologna, Italy.