6

Influence Value Q-Learning: A Reinforcement Learning Algorithm for Multi Agent Systems[1]

Dennis Barrios-Aranibar and Luiz M. G. Gonçalves
Universidade Federal do Rio Grande do Norte
Brazil

1. Introduction

The idea of using agents that can learn to solve problems became popular in the artificial intelligence field, specifically, in machine learning technics. Reinforcement learning (RL) is part of a kind of algorithms called *Reward based learning*. The idea of these algorithms is not to say to the agent what the best response or strategy, but, indicate what the expected result is, thus, the agent must discover what is the best strategy for obtaining the desired result.
Reinforcement learning algorithms calculate a value function for state predicates or for state-action pairs, having as goal the definition of a policy that best take advantage of these values.
Q-learning (Watkins, 1989) is one of the most used reinforcement learning algorithms. It was widely applied in several problems like learning in robotics (Suh et al., 1997; Gu & Hu, 2005), channel assignment in mobile communication systems (Junhong & Haykin, 1999), in the block-pushing problem (Laurent & Piat, 2001), creation of electricity supplier bidding strategies (Xiong et al. 2002), design of intelligent stock trading agents (Lee et al., 2004), design of a dynamic path guidance system based on electronic maps (Zou et al., 2005), mobile robots navigation (Barrios-Aranibar & Alsina, 2004; Tanaka et al., 2007), energy conservation and comfort in buildings (Dalamagkidis et al., 2007), resource allocation (Usaha & Barria, 2007; Vengerov, 2007), and others.
In the other hand, the use of multi-agent systems became popular in the solution of computacional problems like e-commerce (Chen et al., 2008), scheduling in transportation problems (Mes et al., 2007), estimation of energy demand (Toksari, 2007), content based image retrieval (Dimitriadis et al., 2007), between others; and in the solution of problems involving robots like mail sending using robots (Carrascosa et al., 2008), rescue missions (Rooker & Birk, 2005), mapping of structured environments (Rocha et al., 2005), and others.
Also, Q-learning and derived algorithms were applied in multi-agent problems too. For example a fuzzy Q-learning was applied to a multi-player non-cooperative repeated game (Ishibuchi et al., 1997), a hierarchical version of Q-learning (HQL) was applied to learn both the elementary swing and stance movements of individual legs as well as the overall coordination scheme to perform forward movements on a six legged walking machine

[1] This work is supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico CNPq/Brasil

Source: Theory and Novel Applications of Machine Learning, Book edited by: Meng Joo Er and Yi Zhou,
ISBN 978-3-902613-55-4, pp. 376, February 2009, I-Tech, Vienna, Austria

(Kirchner, 1997), a modular Q-learning was applied to multi-agent cooperation in robot soccer (Park et al., 2001), Q-learning was independently applied in a group of agents making economic decisions (Tesauro & Kephart, 2002), and others.
However, when applying Q-Learning to a multi-agent system (MAS), it is important to note that this algorithm was developed for single agent problems. Thus, application of it in MAS problems can be made in several forms. They can be grouped in four paradigms: Team learning, independent learning, joint action learning and influence value learning. The last proposed by authors in previous works (Barrios-Aranibar & Gonçalves, 2007a; Barrios-Aranibar & Gonçalves, 2007b; Barrios-Aranibar & Gonçalves, 2007c).
In this work authors explain the so called IVQ-learning algorithm, which is an extension of the Q-learning algorithm using the concepts of the influence value learning paradigm. In this sense, this chapter is organized as follows: In section 2 we explain the Q-learning algorithm and analyse some extension to it, in section 3 we discuss the four paradigms of application of reinforcement learning in MAS, specially focused in the extensions to Q-learning algorithm, in section 4 we present our algorithm called IVQ-learning and, in section 5 all results of using this algorithm obtained until now are resumed. Finally, conclusions and trends for future works are discussed in section 6.

2. Q-learning

Q-learning is a temporal difference algorithm, where the agent learn independently the action selection policy it is executing. It is important to note that the policy still has an effect in that it determines which state-action pairs are visited and updated. However, all that is required for correct convergence is that all pairs continue to be updated. (Sutton and Barto, 1998). The basic form of the equation for modifying state-action pair value is given by equation 1.

$$Q(s(t),a(t)) \leftarrow Q(s(t),a(t)) + \alpha(r(t+1) + \gamma \max_{a \in A}[Q(s(t+1),a)] - Q(s(t),a(t))) \quad (1)$$

where $Q(s(t),a(t))$ is the value of action $a(t)$ executed by the agent, α is the learning rate ($0 \leq \alpha \leq 1$), γ is the discount rate ($0 \leq \gamma \leq 1$), $r(t+1)$ is the instantaneous reward obtained by the agent and A is the set of actions agent can execute.
In equation 1 can be observed that algorithm updates states action pairs using the maximum of the values of actions of possible next state. The last can be verified in the backup diagram showed in figure 1. Q-learning algorithm is showed in algorithm 1.

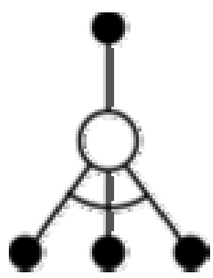


Fig. 1. Backup Diagram of the Q-Learning Algorithm (Source: Sutton & Barto, 1998)

Algorithm 1. Q-learning algorithm

Require: Initialize $Q(s,a)$ with arbitrary values
for all episodes **do**
  Initialize $s(0)$
  $t \leftarrow 0$
  repeat
    Choose action $a(t)$ in state $s(t)$, using a policy derived from Q
    Execute action $a(t)$, observe $r(t+1)$ and $s(t+1)$
    $Q(s(t),a(t)) \leftarrow Q(s(t),a(t)) + \alpha(r(t+1) + \gamma \max_{a \in A}[Q(s(t+1),a)] - Q(s(t),a(t)))$
    $t \leftarrow t+1$
  until $s(t)$ being a terminal state
end for

Watkins and Dayan proved the convergence of this algorithm when the sequence of episodes that forms the basis of learning include an infinite number of episodes for each starting state and action (Watkins & Dayan, 1992). Also, Tsitsiklis proved its convergence over more general conditions (Tsitsiklis, 1994).

Several extensions of this algorithm were proposed. Extensions generally aim to overpass some drawbacks that appear when Q-learning algorithms are applied to specific fields or kind of problems. Some of this extensions include, but are not limited to:

1. The FQ-Learning (Berenji, 1994), which is a Fuzzy Q-Learning algorithm for decision processes in which the goals and/or the constraints, but not necessarily the system under control, are fuzzy in nature.
2. The QLASS algorithm (Murao & Kitamura, 1997), which is a Q-learning algorithm with adaptive state segmentation specially created for learning robots that need to construct a suitable state space without knowledge of the sensor space.
3. The Region-based Q-learning (Suh et al., 1997), which was developed for using in continuous state space applications. the method incorporates a region-based reward assignment being used to solve a structural credit assignment problem and a convex clustering approach to find a region with the same reward attribution property.
4. The Bayesian Q-learning (Dearden et al., 1998), which is a learning algorithm for complex environments that aims to balance exploration of untested actions against exploitation of actions that are known to be good.
5. The kd-Q-learning (Vollbrecht, 2000), which is an algorithm for problems with continuous state space. It approximates the quality function with a hierarchic discretization structure called kd-tree.
6. The SQ-learning (Kamaya et al., 2000), which is memoryless approach for reinforcement learning in partially observable environments.
7. The Continuous-Action Q-Learning (Millán et al., 2002), a Q-learning method that works in continuous domains. It uses an incremental topology preserving map (ITPM) to partition the input space, and incorporates a bias to initialize the learning process.
8. The SA-Q-learning (Maozu et al. 2004), where the Metropolis criterion of simulated annealing algorithm is introduced in order to balance exploration and exploitation of Q-learning.

3. Q-learning in multi agent systems

As explained in section 2, sometimes it is necessary to extend the Q-learning algorithm in order to overpass some problemas that appear when developers try to apply it in some fields. When applying it in multi-agent systems the same occurs.

Specifically, two problems appear, the size of the state space and the convergence capacity of the algorithm. The first problem is related to the fact that agents position in the environment must be part of the state in the system. Thus, when the number of agents increase, the size of the state space increase too and the problem can became computationally intractable. The second problem is related to the fact that Q-learning algorithm and almost all traditional reinforcement learning algorithms were created for problems with one agent, thus, convergence is not assured when these algorithms are applied to MAS problems.

For solving the first problem, three general approaches could be identified: State abstraction, function approximation and hierarchic decomposition (Morales & Sammut, 2004). One example of these efforts for solving this problem is the work of Ono et al., which developed an architecture for modular Q-learning agents, which was designed for reducing each agent's intractably enormous state space caused by the existence of its partners jointly working in the same environment (Ono et al., 1996).

However, the second problem can be considered a critical one. For this reason, this chapter is devoted to it. Two trends can be distinguished: The first one relies in the construction of hybrid algorithms by combining Q-learning or other RL algorithms with other technics like k-neighbours algorithm (Ribeiro et al., 2006) or by combining it with concepts or other fields like pheromone concept from swarm intelligence (Monekosso & Remagnino, 2004).

The pheromone-Q-learning algorithm (phe-Q) deserves especial attention because in this algorithm agents "influence", in certain way, behaviour of other agents. This algorithm was developed to allow agents to communicate and jointly learn to solve a problem (Monekosso and Remagnino, 2004). Equation for modifying state-action pair value for an *i* agent using this algorithm is given by equation 2.

$$\begin{aligned} &Q(s(t),a_i(t)) \leftarrow Q(s(t),a_i(t)) + \alpha(r(t+1) + \\ &\gamma \max_{a_i \in A_i}[Q(s(t+1),a_i) + \xi B(s(t+1),a_i)] - Q(s(t),a_i(t))) \end{aligned} \tag{2}$$

where $Q(s(t),a_i(t))$ is the value of action $a_i(t)$ executed by agent i, α is the learning rate ($0 \le \alpha \le 1$), γ is the discount rate ($0 \le \gamma \le 1$), $r(t+1)$ is the instantaneous reward obtained by agent i, ξ is a sigmoid function of time epochs, and $B(s(t),a_i(t))$ is defined by equation 3

$$B(s(t),a_i(t)) = \frac{\sum_{s \in Na} \Phi(s)}{\sum_{\sigma \in Na} \Phi_{\max}(\sigma)} \tag{3}$$

where $\Phi(s)$ is the pheromone concentration at a point, s in the environment and Na is the set of neighbouring states for a chosen action a. The belief factor is a function of the synthetic pheromone $\Phi(s)$, a scalar value that integrates the basic dynamic nature of the pheromone, namely aggregation, evaporation and diffusion.

The second trend of application of RL algorithms relies in the application of algorithms without combining it with any other technic or concept. Thus, agents will use only the

information existing in the traditional algorithms. As an example, if developers use Q-learning, agents will trust only in Q values and immediate rewards.

In this sense, there exist four paradigms for applying algorithms like Q-learning in multi-agent systems: Team learning, independent learning, joint action learning and influence value reinforcement learning.

The paradigm where agents learn as a team is based in the idea of modelling the team as a single agent. The great advantage of this paradigm is that the algorithms do not need to be modified (For Q-learning implementations, the algorithm 1 is used). But, in robotics and distributed applications, it can be difficult to implement because we need to have a centralized learning process and sensor information processing.

An example of this paradigm using reinforcement learning is the work of Kok and Vlasis (2004) that model the problem of collaboration in multi-agent systems as a Markov Decision Process. The main problem in their work and other similar works is that the applicability becomes impossible when the number of players increases because the number of states and actions increases exponentially.

The problems reported in learning as a team can be solved by implementing the learning algorithms independently in each agent. Thus, in the case of Q-learning, each agent will implement the algorithm 1 without modifications. Several papers show promising results when applying this paradigm (Sen et al., 1994; Kapetanakis & Kudenko, 2002; Tumer et al., 2002). However, Claus and Boutilier (1998) explored the use of independent learners in repetitive games, empirically showing that the proposal is able to achieve only sub-optimal results. The above results are important when analyzed regarding the nature of the used algorithms. It may be noted that the reinforcement learning algorithms aim to take the agent to perform a set of actions that will provide the greatest utility (greater rewards). Below that, in problems involving several agents, it is possible that the combination of optimal individual strategies not necessarily represents an optimal team strategy. In an attempt to solve this problem, many studies have been developed. An example is the one of Kapetanakis & Kudenko (2002) which proposes a new heuristic for computing the reward values for actions based on the frequency that each action has maximum reward. They have shown empirically that their approach converges to an optimal strategy in repetitive games of two agents. Also, they test it in repetitive games with four agents, where, only one agent uses the proposed heuristic, showing that the probability of convergence to optimal strategies increases but is not guaranteed (2004). Another study (Tumer et al., 2002) explores modifications for choosing rewards. The problem of giving correct rewards in independent learning is studied. The proposed algorithm uses collective intelligence concepts for obtaining better results than by applying algorithms without any modification and learning as a team. Even achieving good results in simple problems such as repetitive games or stochastic games with few agents, another problem in this paradigm, which occurs as the number of agents increase, is that traditional algorithms are designed for problems where the environment does not change, that is, the reward is static. However, in multi-agents systems, the rewards may change over time, as the actions of other agents will influence them.

In the current work, although independent learning uses the algorithm 1 without modifications, we will call this algorithm as IQ-Learning.

One way for solving the problem of the independent learning model is learn the best response to the actions of other agents. In this context, the joint action learning paradigm

appears. Each agent should learn what the value of executing their actions in combination with the actions of others (joint action) is. By intuits a model for other agents, it must calculate the best action for actions supposed to be executed by colleagues and/or adversaries (Kapetanakis et al., 2003; Guo et al., 2007). Claus & Bouitilier (1998) explored the use of this paradigm in repetitive games showing that the basic form of this paradigm does not guarantee convergence to optimal solutions. However, the authors indicate that, unlike the independent learning algorithms, this paradigm can be improved if models of other agents are improved.

Other examples include the work of Suematsu and Hayashi that guarantee convergence to optimal solutions (Suematsu & Hayashi, 2002). The work of Banerjee and Sen (Banerjee & Sen, 2007) that proposes a conditional algorithm for learning joint actions, where agents learn the conditional probability of an action be executed by an opponent be optimal. Then, agents use these probabilities for choosing their future actions. The main problem with this paradigm is the number of combinations of states and actions that grows exponentially as the number of states, actions and/or agents grows.

A modified version of the traditional Q-Learning, for joint action learning, the so called JAQ-Learning algorithm (algorithm 2), is defined by the equation 4.

$$\begin{aligned} Q_i(s(t), a1(t), ..., aN(t)) \leftarrow Q_i(s(t), a1(t), ..., aN(t)) + \\ \alpha(r(t+1) + \gamma \max_{a1,...,aN} Q_i(s(t+1), a1, ..., aN) - Q_i(s(t), a1(t), ..., aN(t))) \end{aligned} \quad (4)$$

where ai_t is the action performed by the agent i at time t; N is number of agents, $Q_i(s(t),a1(t),...,aN(t))$ is the value of the joint action $(a1(t),...,aN(t))$ for agent i in the state $s(t)$. $r(t+1)$ is the reward obtained by agent i as it executes action $ai(t)$ and as other agents execute actions $a1(t),...,ai-1(t),ai+1(t),...,aN(t)$ respectively, α is the learning rate $(0 \leq \alpha \leq 1)$, γ is the discount rate $(0 \leq \gamma \leq 1)$.

An agent has to decide between its actions and not between joint actions. For this decision, it uses the expected value of its actions. The expected value includes information about the joint actions and the current beliefs about other agent that is given by (Equation 5):

$$EV(s(t), ai) \leftarrow \sum_{a_{-i} \in A_{-i}} Q(s(t), a_{-i} \cup ai) * \prod_{j \neq i} \Pr_t(a_{-i} j) \quad (5)$$

where ai is an action of agent i, $EV(s(t),ai)$ is the expected value of action ai in state $s(t)$, a_{-i} is a joint action formed only by actions of other agents, A_{-i} is the set of joint actions of other agents excluding agent i, $Q(s(t),a_{-i} \cup ai)$ is the value of a joint action formed by the union of the joint action a_{-i} of all agents excluding i with action ai of agent i in state $s(t)$ and $Pr_t(a_{-i}j)$ is the probability of agent j performs action aj that is part of joint action a_{-i} in state s_t.

Finally, the learning by influence value paradigm proposed by authors in previous work (Barrios-Aranibar & Gonçalves, 2007a; Barrios-Aranibar & Gonçalves, 2007b) is based on the idea of influencing the behaviour of each agent according to the opinion of others. The value of state-action pairs will be a function of the reward of each agent and the opinion that the other players have on the action that the agent execute individually. This opinion should be a function of reward obtained by the agents. That is, if an agent affects the other players pushing their reward below than the previously received, they have a negative opinion for the actions done by the first agent.

Algorithm 2. JAQ-learning algorithm for an agent *i*

Require: Initialize $Q_i(s,a1,...,aN)$ with arbitrary values
for all episodes **do**
Initialize *s(0)* (the same initial state for all agents)
$t \leftarrow 0$
repeat

$$EV_i(s(\mathrm{t}), ai) \leftarrow \sum_{a\_i \in A\_i} Q_i(s(\mathrm{t}), a\_i \cup ai) * \prod_{j \neq i} \Pr_i(a\_i j)$$

Choose action *ai(t)* in state *s(t)*, using a policy derived from EV_i
Execute action *ai(t)*, observe *r(t +1)* and *s(t +1)*
Observe actions of all agents in the system (*a1(t),...,ai-1(t),ai+1(t),...,aN(t)*)

$$Q_i(s(t), a1(t), ..., aN(t)) \leftarrow Q_i(s(t), a1(t), ..., aN(t)) +$$
$$\alpha(r(t+1) + \gamma \max_{a1,...,aN} Q_i(s(t+1), a1, ..., aN) - Q_i(s(t), a1(t), ..., aN(t)))$$

$t \leftarrow t + 1$
until *s(t)* being a terminal state
end for

From the theoretical point of view, the model proposed does not have the problems related to the paradigms of team learning and joint action learning about the number of actors, actions and states. Finally, when talking about possible changes of rewards during the learning process and that the agent must be aware that the rewards may change because of the existence of other agents, authors conjecture that this does not represent a problem for this paradigm, based on experiments conducted until now.
This paradigm is based on social interactions of people. Some theories about the social interaction can be seen in theoretical work in the area of education and psychology, such as the work of Levi Vygotsky (Oliveira & Bazzan, 2006; Jars et al., 2004).
Based on these preliminary studies on the influence of social interactions in learning, we conjecture that when people interact, they communicate each other what they think about the actions of the other, either through direct criticism or praise. This means that if person A does not like the action executed by the person B, then A will protest against B. If the person B continue to perform the same action, then A can become angry and protest angrily. We abstract this phenomenon and determined that the strength of protests may be proportional to the number of times the bad action is repeated.
Moreover, if person A likes the action executed by the person B, then A must praise B. Even if the action performed is considered as very good, then A must praise B vehemently. But if B continues to perform the same action, then A will get accustomed, and over time it will cease to praise B. The former can be abstracted by making the power of praise to be inversely proportional to the number of times the good action is repeated.
More importantly, we observe that protests and praises of others can influence the behaviour of a person. Therefore, when other people protest, a person tries to avoid actions that caused these protests and when other people praise, a person tries to repeat actions more times.

4. IVQ-learning

Inspired in the facts explained before, in influence value paradigm, agents estimate the values of their actions based on the reward obtained and a numerical value called influence value. The influence value for an agent *i* in a group of *N* agents is defined by equation 6.

$$IV_i \leftarrow \sum_{j\in(1:N),j\neq i} \beta_i(j) * OP_j(i) \tag{6}$$

Where $\beta_i(j)$ is the influence rate of agent j over agent i, $OP_j(i)$ is the opinion of agent j about the action executed by agent i.

The influence rate β determine whether or not an agent will be influenced by opinions of other agents. OP is a numerical value which is calculated on the basis of the rewards that an agent has been received. Because in reinforcement learning the value of states or state-action pairs is directly related to rewards obtained in the past, then the opinion of an agent will be calculated according to this value and reward obtained at the time of evaluation (Equation 7).

$$OP_j(i) \leftarrow \begin{cases} RV_j * Pe(s(t), a_i(t)) & If \quad RV_j \leq 0 \\ RV_j * (1 - Pe(s(t), a_i(t))) & In \quad other \quad case \end{cases} \tag{7}$$

Where

$$RV_j \leftarrow r_j + \max_{a_j \in A_j} Q(s(t+1), a_j) - Q(s(t), a_j(t)) \tag{8}$$

For the case to be learning the values of state-action pairs. $Pe(s(t),a_i(t))$ is the occurrence index (times action a_i is executed by agent i in state $s(t)$ over times agent i have been in state $s(t)$). $Q(s(t),a_j(t))$ is the value of the state-action pair of the agent j at time t. And, A_j is the set of all actions agent j can execute. Thus, in the IVQ-learning algorithm based on Q-Learning, the state-action pair value for an agent i is modified using the equation 9.

$$\begin{aligned} Q(s(t), a_i(t)) \leftarrow Q(s(t), a_i(t)) + \alpha(r(t+1) + \\ \gamma \max_{a_i \in A_i} Q(s(t+1), a_i) - Q(s(t), a_i(t)) + IV_i) \end{aligned} \tag{9}$$

where $Q(s(t),a_i(t))$ is the value of action $a_i(t)$ executed by agent i, α is the learning rate ($0 \leq \alpha \leq 1$), γ is the discount rate ($0 \leq \gamma \leq 1$).And, $r(t+1)$ is the instantaneous reward obtained by agent i.

In this sense, the IVQ-Learning algorithm that extends the Q-learning algorithm by using equations 6 to 9 is presented in algorithm 3.

Algorithm 3. IVQ-learning algorithm for an agent i

Require: Initialize $Q(s,ai)$ with arbitrary values
for all episodes **do**
  Initialize $s(0)$ (the same initial state for all agents)
  $t \leftarrow 0$
  repeat
    Choose action $ai(t)$ in state $s(t)$, using a policy derived from Q
    Execute action $ai(t)$, observe $r(t+1)$ and $s(t+1)$
    $RV_i \leftarrow r_i + \max_{a_i \in A_i} Q(s(t+1), a_i) - Q(s(t), a_i(t))$
    Observe actions of all agents in the system ($a1(t),...,ai\text{-}1(t),ai\text{+}1(t),...,aN(t)$)
    for j = all agents except i **do**

$$OP_i(j) \leftarrow \begin{cases} RV_i * Pe(s(t), a_j(t)) & If \quad RV_i \leq 0 \\ RV_i * (1 - Pe(s(t), a_j(t))) & In \quad other \quad case \end{cases}$$

end for
Observe opinions of all agents in the system ($OP_1(i)$,..., $OP_{i-1}(i)$, $OP_{i+1}(i)$,..., $OP_N(i)$)

$$IV_i \leftarrow \sum_{j\in(1:N),j\neq i} \beta_i(j) * OP_j(i)$$

$$Q(s(t),a_i(t)) \leftarrow Q(s(t),a_i(t)) + \alpha(r(t+1) + \gamma \max_{a_i \in A_i} Q(s(t+1),a_i) - Q(s(t),a_i(t)) + IV_i)$$

$t \leftarrow t+1$
until $s(t)$ being a terminal state
End for

5. Experimental results

This section summarizes results obtained by using IVQ-learning algorithm in comparison with the IQ-learning ang JAQ-Learning algorithms (Barrios-Aranibar & Gonçalves, 2007a; Barrios-Aranibar & Gonçalves, 2007b; Barrios-Aranibar & Gonçalves, 2007c).
Before talking about our results, it is important to know that when talking about cooperative agents or robots, it is necessary that agents cooperate on equality and that all agents receive equitable rewards for solving the task. Is in this context that a different concept from the games theory appears in multi-agent systems. This is the concept of the Nash equilibrium.
Let be a multi-agent system formed by N agents. σ^*_i is defined as the strategy chosen by the agent i, σ_i as any strategy of the agent i, and Σ_i as the set of all possible strategies of i. It is said that the strategies $\sigma^*_1,..., \sigma^*_N$ constitute a Nash equilibrium, if inequality 10 is true for all $\sigma_i \in \Sigma_i$ and for all agents i.

$$r_i(\sigma^*_1,...,\sigma^*_{i-1},\sigma_i,\sigma^*_{i+1},...,\sigma^*_N) \leq r_i(\sigma^*_1,...,\sigma^*_{i-1},\sigma^*_i,\sigma^*_{i+1},...,\sigma^*_N) \quad (10)$$

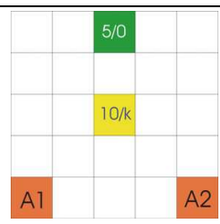
Where r_i is the reward obtained by agent i.
The idea of Nash equilibrium, is that the strategy of each agent is the best response to the strategies of their colleagues and/or opponents (Kononen, 2004). Then, it is expected that learning algorithms can converge to a Nash equilibrium, and it is desired that can converge to the optimal Nash equilibrium, that is the one where the reward for all agents is the best.
We test and compare all paradigms using two repetitive games (The penalty problem and the climbing problem) (Barrios-Aranibar & Gonçalves, 2007a) and one stochastic game for two agents (Barrios-Aranibar & Gonçalves, 2007b). The penalty problem, in which IQ-Learning, JAQ-Learning and IVQ-Learning can converge to the optimal equilibrium over certain conditions, is used for testing capability of those algorithms to converge to optimal equilibrium. And, the climbing problem, in which IQ-Learning, JAQ-Learning can not converge to optimal equilibrium was used to test if IVQ-Learning can do it. Also, a game called the grid world game was created for testing coordination between two agents. Here, both agents have to coordinate their actions in order to obtain positive rewards. Lack of coordination causes penalties. Figure 2 shows the three games used until now.
In penalty game, $k < 0$ is a penalty. In this game, there exist three Nash equilibriums ((a0,b0), (a1,b1) and (a2,b2)), but only two of them are optimal Nash equilibrums ((a0, b0) and (a3, b3)). When $k = 0$ (no penalty for any action in the game), the three algorithms (IQ-Learning, JAQ-Learning and IVQ-Learning) converge to the optimal equilibrium with probability one. However, as k decrease, this probability also decrease.

	a0	a1	a2
b0	10	0	k
b1	0	2	0
b2	k	0	10

(a)

	a0	a1	a2
b0	11	-30	0
b1	-30	7	6
b2	0	0	5

(b)



(c)

Fig. 2. Games used for testing performance of paradigms for applying reinforcement learning in multi-agent systems: (a) Penalty game, (b) Climbing game, (c) Grid world game.

Figure 3 compiles results obtained by these three algorithms in the penalty game, all of them was executed with the same conditions: A Boltzman action selection strategy with initial temperature T = 16, λ = 0.1 and in the case of IVQ-Learning β = 0.05. Also, a varying decaying rate for T was defined and each algorithm was executed 100 times for each decaying rate.

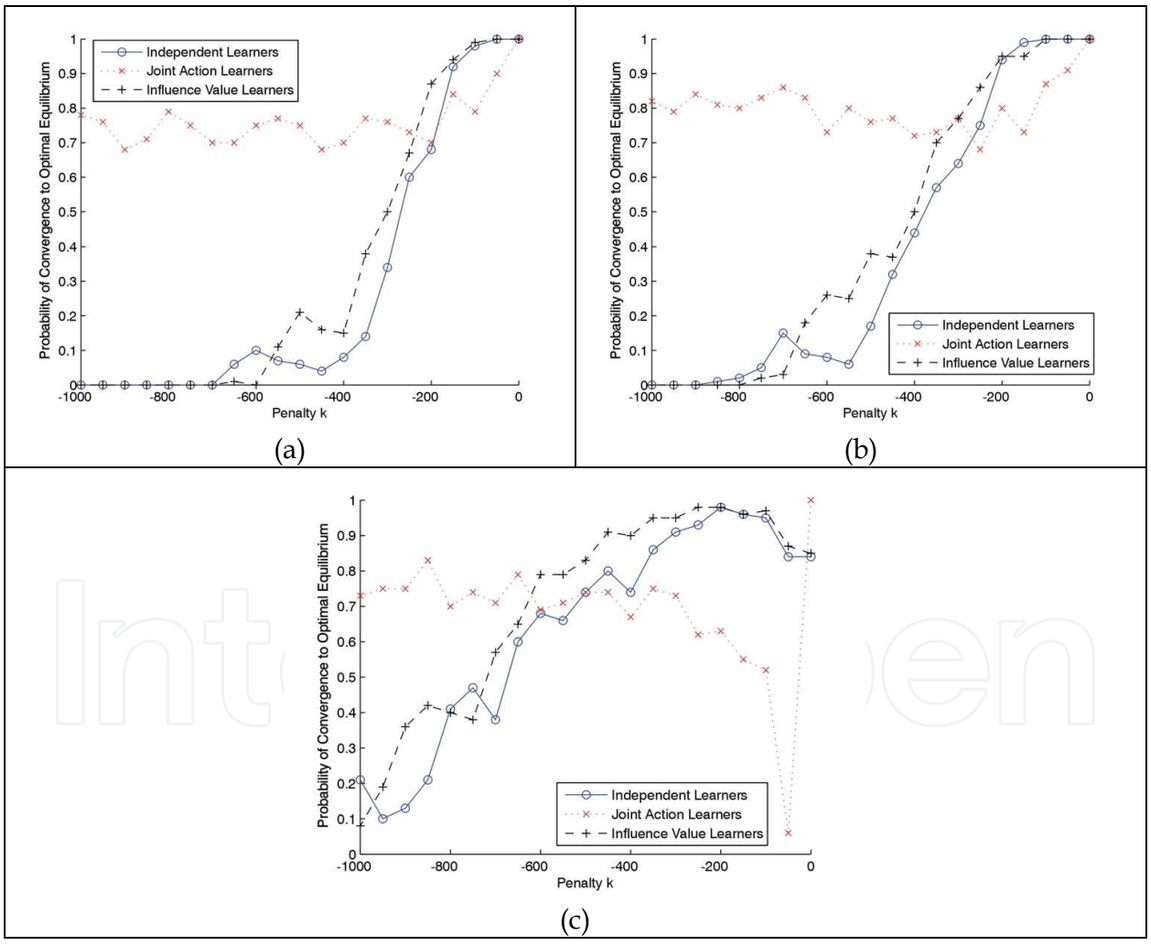


Fig. 3. Probability of convergence to optimal equilibrium in the penalty game for λ = 0.1, β = 0.05 and (a) $T = 0.998^t * 16$, (b) $T = 0.999^t * 16$, and (c) $T = 0.9999^t * 16$.

In this problem JAQ-Learning has the best perform. But, it is important to note also that for values of k near to zero, IVQ-Learning and IQ-Learning performs better than the JAQ-

Learning, and for those values the IVQ-Learning algorithm has the best probability to converge to the optimal equilibrium.

The climbing game problem is specially difficult for reinforcement learning algorithms because action a2 has the maximum total reward for agent A and action b1 has the maximum total reward for agent B. Independent learning approaches and joint action learning was showed to converge in the best case only to the (a1, b1) action pair (Claus and Boutilier, 1998). Again, each algorithm was executed 100 times in the same conditions: A Boltzman action selection strategy with initial temperature T = 16, $\lambda = 0.1$ and in the case of IVQ-Learning $\beta = 0.1$ and a varying temperature decaying rate.

In relation to the IQ-Learning and the JAQ-Learning, obtained results confirm that these two algorithms can not converge to optimal equilibrium. IVQ-Learning is the unique algorithm that has a probability different to zero for converging to the optimal Nash equilibrium, but this probability depends on the temperature decaying rate of the Boltzman action selection strategy (figure 4). In experiments, the best temperature decaying rate founded was 0.9997 on which probability to convergence to optimal equilibrium (a0, b0) is near to 0.7.

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

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.

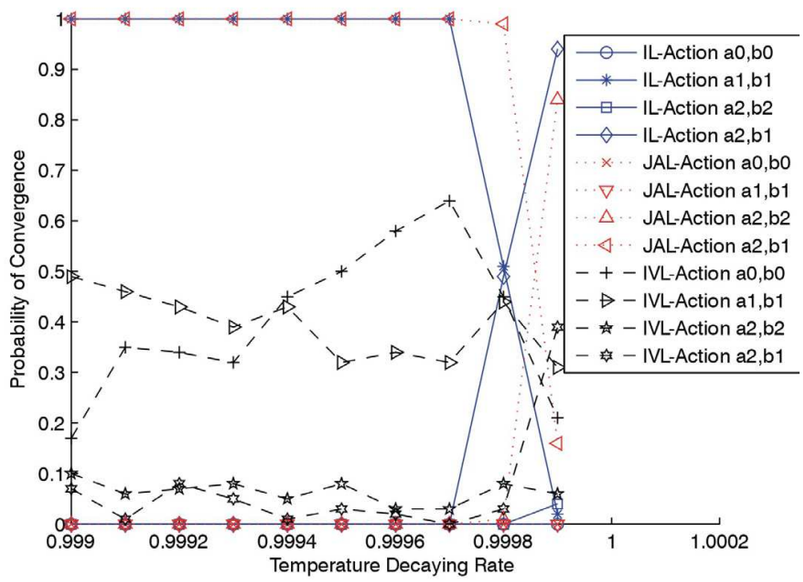


Fig. 4. Probability of Convergence in Climbing Game with $\lambda = 0.1$, $\beta = 0.1$ and Variable Temperature Decaying Rate

The first tested algorithm (Independent Learning A) considers that the state for each agent is the position of the agent, thus, the state space does not consider the position of the other agent. The second version of this algorithm (Independent Learning B) considers that the state space is the position of both agents. The third one is the JAQ-Learning algorithm and the last one is the IVQ-Learning.

In the tests, each learning algorithm was 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.

Figure 5 shows the probability of reaching the position (3; 3) with α=1, λ=0.1, β=0:1 and T = 0:99t. In this figure, 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. 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.

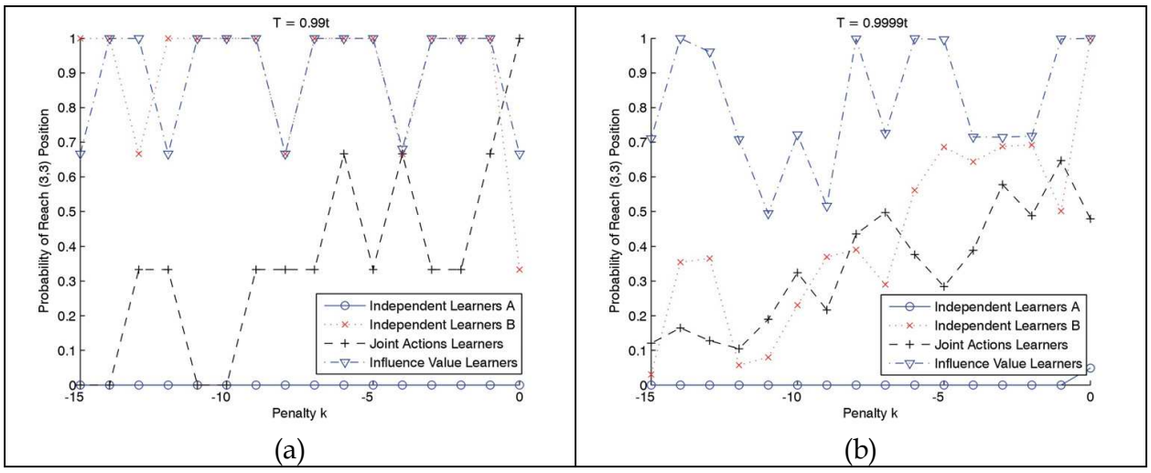


Fig. 5. Probability of reaching (3,3) position for (a) T = 0.99t and (b) T = 0.9999t

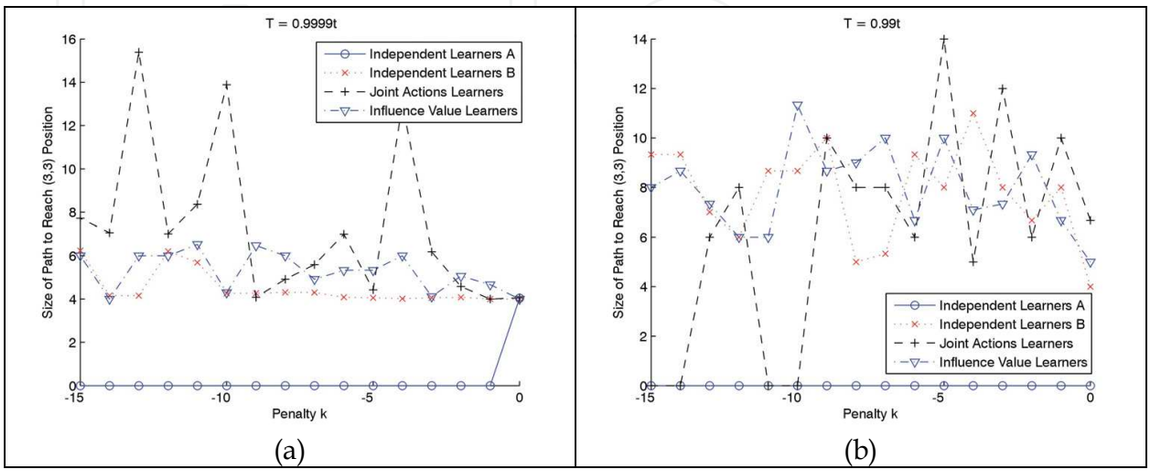


Fig. 6. Size of path for reaching (3,3) position for (a) T = 0.99t and (b) T = 0.9999t

As shown in figure 6, as more exploratory the action selection policy is, smaller is the size of the path for reaching (3; 3) position. Then, it can concluded that 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. It can be concluded by joining the probability of convergence to the position (3; 3) and the mean size of the path for reaching this position.

In order to test collaboration and self organization (automatic task assignment) in a group of reinforcement learning agents, authors created the foraging game showed in figure 7 (Barrios-Aranibar and Gonçalves, 2007c). In this game, a team of agents have to find food in the environment and eat it. When food in the environment no more exists, then, the game finishes. Initially, agents do not know that by reaching food they are going to win the game, then, they have to learn that eat food is good for them and also they have to learn to find it in the environment in order to win the game.

IQ-Learning, JAQ-Learning and IVQ-Learning were implemented in this problem with *20000* learning epochs. Also each algorithm was trained 10 times, and 3 different values of parameter α (*0.05*, *0.1*, *0.15*) were used. Because our approach (IVQ-Learning) has an extra parameter (β), it was trained with six different values: β={*0.05*, *0.1*, *0.15*, *0.2*, *0.25*, *0.3*}. Thus, we constructed 8 algorithms and trained it 10 times each one. For all algorithms, the parameter γ was chosen to be *0.05*.

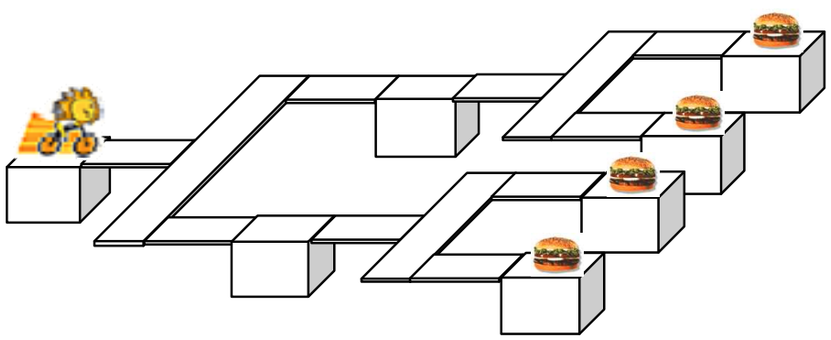
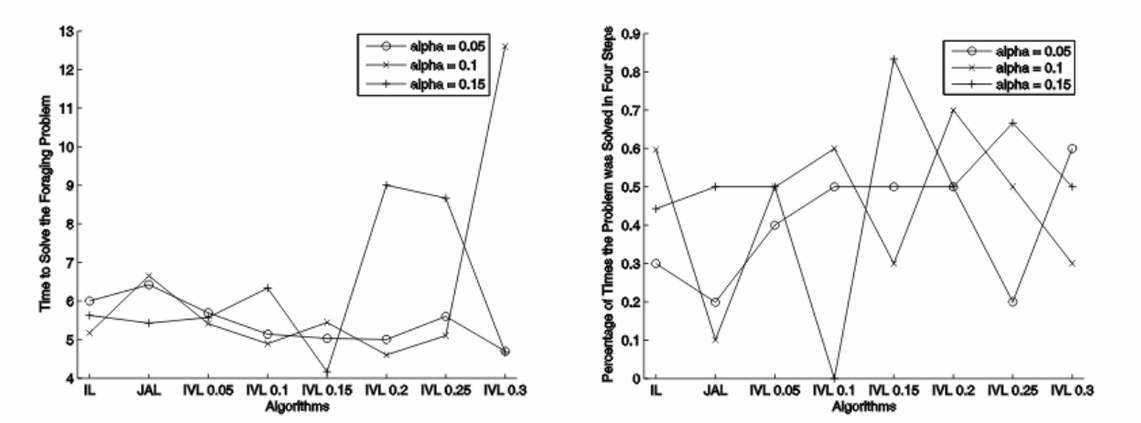


Fig. 7. Foraging game for testing collaboration between agents.

In figure 8.a, a comparison of these eight algorithms is showed. This comparison is based on the number of steps needed by the two agents to solve the problem. This value is calculated considering the mean of *100* tests for each algorithm and parameter α. In the optimal strategy the number of steps must be four. In this context, it was observed that our approach with parameters β=*0.15* and α=*0.15* had the best performance.

Figure 8.a shows the mean of number of steps need for each algorithm to solve the problem. But, in certain tests, the algorithms could converge to the optimal strategy (four steps). Then it is important to analyze the number of times each algorithm converge to this strategy. This analysis is showed in figure 8.b. In this figure, the percentage of times each algorithm converge to the optimal solution is showed. Again, it could be observed that our approach performs better. Also, the best IVQ-Learning was the one with parameters β=*0.15* and α=*0.15*.



(a) (b)

Fig. 8. Comparison between three paradigms when applyied to the foraging game.

6. Conclusion and future works

In this work, we explain the extension made to the Q-learning algorithm by using the influence value reinforcement learning paradigm. Also, we present a summary of all results obtained by comparing our approach with the IQ-learning and JAQ-learning algorithms. After analyse these results it is possible to note that our approach had advantages over the traditional paradigms and encourage authors to continue researching in this paradigm.

Also, our paradigm is an intend to solve the problem of convergence generated when applying Q-learning in multi-agent systems but in future works it is necessary to explore solutions for the problem related to the size of the state space.

7. References

Banerjee, D. and Sen, S. (2007). Reaching pareto-optimality in prisoner's dilemma using conditional joint action learning. *Autonomous Agents and Multi-Agent Systems* 15(1), 91–108.

Barrios-Aranibar, D. and Alsina, P. J. (2004), Reinforcement Learning-Based Path Planning for Autonomous Robots. In *I ENRI - Encontro Nacional de Robótica Inteligente no XXIV Congresso da Sociedade Brasileira de Computação*, Salvador, BA, Brazil, 08/2004.

Barrios-Aranibar, D. and Gonçalves, L. M. G. (2007a). Learning Coordination in Multi-Agent Systems using Influence Value Reinforcement Learning. In: *7th International Conference on Intelligent Systems Design and Applications (ISDA 07)*, 2007, Rio de Janeiro. Pages: 471-478.

Barrios-Aranibar, D. and Gonçalves, L. M. G. (2007b). Learning to Reach Optimal Equilibrium by Influence of Other Agents Opinion. In: *Hybrid Intelligent Systems, 2007. HIS 2007. 7th International Conference on*, 2007, Kaiserslautern. pp. 198-203.

Barrios-Aranibar, D. and Gonçalves, L. M. G. (2007c). Learning to Collaborate from Delayed Rewards in Foraging Like Environments. In: *VI Jornadas Peruanas De Computación - JPC 2007*.

Berenji, H. R. (1994), Fuzzy Q-learning: a new approach for fuzzy dynamic programming. In *Fuzzy Systems, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the Third IEEE Conference on*. 26-29 June 1994. 1:486-491.

Carrascosa, C.; Bajo, J.; Julian, V.; Corchado, J. M. and Botti, V. (2008). Hybrid multi-agent architecture as a real-time problem-solving model. *Expert Systems with Applications*. 34(1):2–17.

Chen, D.; Jeng, B.; Lee, W. and Chuang, C. (2008). An agent-based model for consumer-to-business electronic commerce. *Expert Systems with Applications*. 34(1): 469–481.

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

Dalamagkidis, K.; Kolokotsa, D.; Kalaitzakis, K. and Stavrakakis, G. S. (2007), Reinforcement learning for energy conservation and comfort in buildings. *Building and Environment*. 42(7):2686–2698.

Dearden, R.; Friedman, N. and Russell, S. (1998). Bayesian Q-learning. In *(1998) Proceedings of the National Conference on Artificial Intelligence*. 761-768.

Dimitriadis, S.; Marias, K. and Orphanoudakis, S. C. (2007). A multi-agent platform for content-based image retrieval. *Multimedia Tools and Applications*. 33(1):57–72.

Gu D. and Hu H. (2005). Teaching robots to plan through Q-learning. *Robotica*. 23: 139-147 Cambridge University Press.

Guo, R., Wu, M., Peng, J., Peng, J. and Cao, W. (2007). New q learning algorithm for multi-agent systems, *Zidonghua Xuebao/Acta Automatica Sinica*, 33(4), 367–372.

Ishibuchi, H.; Nakashima, T.; Miyamoto, H. and Chi-Hyon Oh (1997), Fuzzy Q-learning for a multi-player non-cooperative repeated game. In *Fuzzy Systems, 1997., Proceedings of the Sixth IEEE International Conference on*. 1-5 July 1997. 3:1573-1579.

Jars, I., Kabachi, N. and Lamure, M. (2004). Proposal for a vygotsky's theory based approach for learning in MAS. In *AOTP: The AAAI-04 Workshop on Agent Organizations: Theory and Practice*. San Jose, California. http://www.cs.uu.nl/virginia/aotp/papers/AOTP04IJars.Pdf.

Junhong N. and Haykin, S. (1999), A Q-learning-based dynamic channel assignment technique for mobile communication systems. *Vehicular Technology, IEEE Transactions on*. 48(5):1676-1687. Sept. 1999.

Kamaya, H.; Lee, H. and Abe, K. (2000), Switching Q-learning in partially observable Markovian environments. In *Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*. 31 Oct.-5 Nov. 2000, 2:1062-1067.

Kapetanakis, S. and Kudenko, D. (2002). Reinforcement learning of coordination in cooperative multi-agent systems. In *Proceedings of the National Conference on Artificial Intelligence*. pp. 326–331.

Kapetanakis, S. and Kudenko, D. (2004). Reinforcement learning of coordination in heterogeneous cooperative multi-agent systems. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2004*. Vol. 3, pp. 1258–1259.

Kapetanakis, S., Kudenko, D. and Strens, M. J. A. (2003). Reinforcement learning approaches to coordination in cooperative multi-agent systems. *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science) 2636*, 18–32.

Kirchner, F. (1997), Q-learning of complex behaviours on a six-legged walking machine. In *Advanced Mobile Robots, 1997. Proceedings., Second EUROMICRO workshop on*. 22-24 Oct. 1997. 51 – 58.

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

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

Laurent, G. and Piat, E. (2001), Parallel Q-learning for a block-pushing problem. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*. 29 Oct.-3 Nov. 2001. 1:286-291.

Lee, J.W.; Hong, E. and Park, J.(2004), A Q-learning based approach to design of intelligent stock trading agents. In *Engineering Management Conference, 2004. Proceedings. 2004 IEEE International*. 18-21 Oct. 2004. 3:1289–1292.

Mes, M.; van der Heijden, M. and van Harten, A. (2007). Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems. *European Journal of Operational Research*. 181(1):59–75.

Maozu G.; Yang L. and Malec, J. (2004), A new Q-learning algorithm based on the metropolis criterion. *Systems, Man, and Cybernetics, Part B, IEEE Transactions on*. Oct. 2004. 34(5):2140-2143.

Millán, J. R.; Posenato, D. and Dedieu, E. (2002), Continuous-Action Q-Learning. *Machine Learning*. November, 2002. 49(2-3): 247-265.

Monekosso, N. and Remagnino, P. (2004), The analysis and performance evaluation of the pheromone-Q-learning algorithm. *Expert Systems* 21(2):80-91.

Morales , E. F. and Sammut, C. (2004), Learning to fly by combining reinforcement learning with behavioural clonning. In *Twenty-first international conference on Machine Learning,* ACM International Conference Proceeding Series. ACM Pres New York, NY, USA.

Murao, H.; Kitamura, S. (1997), Q-Learning with adaptive state segmentation (QLASS). In *Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings., 1997 IEEE International Symposium on*. 10-11 July 1997. 179 – 184.

Oliveira, D. De and Bazzan, A. L. C. (2006). Traffic lights control with adaptive group formation based on swarm intelligence. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 4150 LNCS*, 520–521.

Ono, N.; Ikeda, O. and Fukumoto, K. (1996), Acquisition of coordinated behavior by modular Q-learning agents. In *Intelligent Robots and Systems '96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*. 4-8 Nov. 1996. 3:1525-1529.

Park, K. H.; Kim, Y. J. and Kim, J. H. (2001), Modular Q-learning based multi-agent cooperation for robot soccer, *Robotics and Autonomous Systems*. 31 May 2001, 5(2): 109-122.

Ribeiro, R.; Enembreck, F. and Koerich, A. L. (2006). A hybrid learning strategy for discovery of policies of action. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 4140 LNAI: 268–277.

Rocha, R.; Dias, J. and Carvalho, A. (2005). Cooperative multi-robot systems: A study of vision-based 3-d mapping using information theory. *Robotics and Autonomous Systems*. 53(3-4):282–311.

Rooker, M. N. and Birk, A. (2005). Combining exploration and ad-hoc networking in robocup rescue. *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*. 3276:236–246.

Sen, S., Sekaran, M. and Hale, J. (1994). Learning to coordinate without sharing information. In *Proceedings of the National Conference on Artificial Intelligence*. Vol. 1, pp. 426–431.

Suematsu, N. and Hayashi, A. (2002), A multiagent reinforcement learning algorithm using extended optimal response. In *Proceedings of the International Conference on Autonomous Agents*. number 2, pp. 370–377.

Suh, I. H.; Kim, J. H. and Oh, S. R. (1997), Region-based Q-learning for intelligent robot systems. In *Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings., 1997 IEEE International Symposium on*. 10-11 July 1997. 172-178.

Sutton, R. and Barto, A. (1998), *Reinforcement learning: an introduction*. MIT Press, Cambridge, MA, 1998.

Tanaka, T.; Nishida, K. and Kurita, T. (2007). Navigation of mobile robot using location map of place cells and reinforcement learning. *Systems and Computers in Japan*. 38(7), 65–75.

Tesauro, G. and Kephart, J. O. (2002), Pricing in Agent Economies Using Multi-Agent Q-Learning. *Autonomous Agents and Multi-Agent Systems*. September, 2002. 5(3): 289-304.

Toksari, M. D. (2007). Ant colony optimization approach to estimate energy demand of turkey. *Energy Policy*. 35(8):3984–3990.

Tsitsiklis, J. N. (1994), Asynchronous Stochastic Approximation and Q-Learning. *Machine Learning*. Kluwer Academic Publishers, Boston. 16(3): 185-202.

Tumer, K., Agogino, A. K. and Wolpert, D. H. (2002). Learning sequences of actions in collectives of autonomous agents. In *Proceedings of the International Conference on Autonomous Agents*. número 2, pp. 378–385.

Usaha, W. and Barria, J. A. (2007), Reinforcement learning for resource allocation in leo satellite networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*. 37(3): 515–527.

Vengerov, D. (2007), A reinforcement learning approach to dynamic resource allocation. *Engineering Applications of Artificial Intelligence*. 20(3): 383–390.

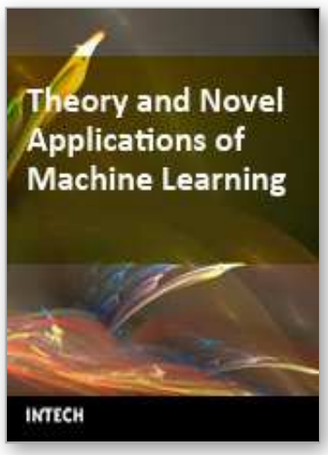
Vollbrecht, H. (2000), Hierarchic function approximation in kd-Q-learning. In *Knowledge-Based Intelligent Engineering Systems and Allied Technologies, 2000. Proceedings. Fourth International Conference on*. 30 Aug.-1 Sept. 2000, 2:466-469.

Watkins, C. J. C. H. (1989), *Learning from Delayed Rewards*. PhD thesis, Cambridge University, Cambridge, England.

Watkins, C. J. C. H. and Dayan, P. (1992), Q-Learning. *Machine Learning*. Kluwer Academic Publishers, Boston. 8(3-4): 279-292.

Xiong, G.; Hashiyama, T. and Okuma, S. (2002), An electricity supplier bidding strategy through Q-learning. In *(2002) Proceedings of the IEEE Power Engineering Society Transmission and Distribution Conference*. 3 (SUMMER): 1516-1521.

Zou L.; Xu, J. and Zhu, L. (2005). Designing a dynamic path guidance system based on electronic maps by using Q-learning. In *Proc. SPIE*. 5985, 59855A. International Conference on Space Information Technology. DOI:10.1117/12.658569,.



Theory and Novel Applications of Machine Learning

Edited by Meng Joo Er and Yi Zhou

ISBN 978-953-7619-55-4
Hard cover, 376 pages
Publisher InTech
Published online 01, January, 2009
Published in print edition January, 2009

Even since computers were invented, many researchers have been trying to understand how human beings learn and many interesting paradigms and approaches towards emulating human learning abilities have been proposed. The ability of learning is one of the central features of human intelligence, which makes it an important ingredient in both traditional Artificial Intelligence (AI) and emerging Cognitive Science. Machine Learning (ML) draws upon ideas from a diverse set of disciplines, including AI, Probability and Statistics, Computational Complexity, Information Theory, Psychology and Neurobiology, Control Theory and Philosophy. ML involves broad topics including Fuzzy Logic, Neural Networks (NNs), Evolutionary Algorithms (EAs), Probability and Statistics, Decision Trees, etc. Real-world applications of ML are widespread such as Pattern Recognition, Data Mining, Gaming, Bio-science, Telecommunications, Control and Robotics applications. This books reports the latest developments and futuristic trends in ML.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen