Fuzzy Q-Learning in Nonstationary Environments

Martin Appl, Dr. Rainer Palm
Siemens AG, Corporate Technology
Information and Communications
D-81730 Munich, Germany
Phone: +49-89-636-45698 Fax: +49-89-636-45456
email: {Martin.Appl, Rainer.Palm}@mchp.siemens.de

ABSTRACT: Classical reinforcement learning approaches, like the well known Q-learning algorithm proposed by Watkins and its fuzzy extensions, presuppose that the characteristics of the environment in which learning takes place are time-independent. This assumption does not hold in many practical applications e.g. due to the aging or failure of components of the environment. In this paper an extension of fuzzy Q-learning, *characterization and clustering of reinforcement signals (CCR)*, is introduced, enabling agents to learn in environments with discretely and continuously changing characteristics. The learning agents are capable of tracking the characteristics of the environment in order to reuse knowledge on future returns of known situations. Moreover, suitable strategies for unknown situations can be deduced by the agents from prior experiences. Finally, the use of a hierarchical clustering algorithm for the aggregation of redundant knowledge is proposed. The soundness of the CCR algorithm is shown by examples from signal optimization in urban traffic networks.

KEYWORDS: reinforcement learning, dynamic programming, nonstationary environment, characterization and clustering of reinforcement signals, CCR, fuzzy Q-learning, traffic signal control

INTRODUCTION

Reinforcement learning deals with learning to act in an optimal way based on the interaction with the environment. One popular approach is the Q-learning algorithm proposed by Watkins (1989), which enables agents to learn optimal strategies without building an explicit model of their environment. Fuzzy Q-learning (Glorennec 1997, Ishibuchi 1997) extends Q-learning to continuous state-action spaces. Both ordinary Q-learning and fuzzy Q-learning are restricted to stationary environments, e.g. environments returning feedback according to time-independent probability distributions. Due to hidden states this restriction is not valid in many practical applications, like the optimization of traffic signals proposed in this paper. In such nonstationary applications learning agents are repeatedly forced to adapt their strategies in order to achieve maximum future rewards.
The new method proposed in this paper, *characterization and clustering of reinforcement signals (CCR)*, enables agents to learn in continuously or discretely changing environments. Knowledge about the different characteristics of the environment and the appropriate optimal strategies is preserved, and reasonable strategies for unknown situations may be deduced from prior experiences. The amount of memory required to preserve this knowledge may be reduced by the clustering of similar characteristics. No restrictions on the dynamics of the environment, i.e. the hidden states, are made and no a priori knowledge about the dynamics is needed. The remainder of this paper is organized as follows. The succeeding section gives a short introduction to the notation used in this paper and to fuzzy Q-learning. Afterwards, the new CCR algorithm will be described in detail. To show the soundness of the proposed method an example from traffic signal control will finally be given.

FUZZY Q-LEARNING

A typical framework for reinforcement learning is shown in figure 1. A learning agent has to select actions a based on the states i of the environment and receives reinforcement signals g according to the suitability of the selected actions. The agent's goal is to find a policy, i.e. a mapping p from states i to actions a, such that the expected discounted sum of future rewards, J^p, is maximized:

$$J^p(i_k) = \mathrm{E}\left\{\sum_{k=0}^{\infty} a^k\, g\big(i_{k+k}, p(i_{k+k}), i_{k+k+1}\big) \,\middle|\, i_k\right\}, \tag{1}$$

where i_k is the state in time step k, $g\big(i_{k+k}, p(i_{k+k}), i_{k+k+1}\big)$ is the reward for performing action $p(i_{k+k})$ in state i_{k+k}, and a ($0 < a < 1$) is a discount factor. The environment is supposed to transit on action $p(i_{k+k})$ from state i_{k+k} to state i_{k+k+1} with probability $p(i_{k+k}, p(i_{k+k}), i_{k+k+1})$.

In Q-learning, the agent learns an evaluation function called "Q-function". $Q(i,a)$ gives the expected discounted sum of future rewards on performing action a in state i and behaving optimally afterwards. The Q-function for a n-dimensional state-space is represented by a fuzzy inference system with m rules of the form

if $i_{k,1}$ is $I^{(1)}_{l_{j,1}}$ **and** $i_{k,2}$ is $I^{(2)}_{l_{j,2}}$ **and** ... **and** $i_{k,n}$ is $I^{(n)}_{l_{j,n}}$ **and** a_k is $A_{\bar{l}_j}$ **then** $Q_k(i_k,a_k)=q_k(I_{l_j}, A_{\bar{l}_j})$, $j=1,...,m$ (2)

with fuzzy labels $I^{(c)}_{l_{j,c}}$ and $A_{\bar{l}_j}$ for the components $c=1,...,n$ of the state and the action of rule j, $j=1,...,m$. The total output of the inference system is given by the weighted sum of the outputs of the single rules

$$Q_k(i,a) = \sum_{j=1}^{m} m_j q_k(I_{l_j}, A_{\bar{l}_j}), \tag{3}$$

where it is assumed that the sum of the degrees of membership m_j=($i_{k,1}$ is $I^{(1)}_{l_{j,1}}$ **and** $i_{k,2}$ is $I^{(2)}_{l_{j,2}}$ **and** ... **and** $i_{k,n}$ is $I^{(n)}_{l_{j,n}}$ **and** a_k is $A_{\bar{l}_j}$) equals 1 for each (i,a).

At each time step, the learning agent selects an action according to the state of the environment and the learned Q-function. If the correct Q-function were known, the agent should, by definition, perform in each state the action that maximizes the Q-function. Since this is not the case during learning, the Q-function is merely used as a bias for the action selection. Having executed action a_k and receiving reward $g\big(i_k, a_k, i_{k+1}\big)$ at time step k, the Q-values in the consequents of the rules (2) are updated according to the activation m_j of the corresponding rules:

$$q_{k+1}(I_{l_j}, A_{\bar{l}_j}) = q_k(I_{l_j}, A_{\bar{l}_j}) + h_k m_j \Big(g(i_k, a_k, i_{k+1}) + a V_k(i_{k+1}) - q_k(I_{l_j}, A_{\bar{l}_j}) \Big), \tag{4}$$

where h_k is a step size that should tend to zero as k increases, and $V_k(i_{k+1}) = \max_a Q_k(i_{k+1}, a)$ gives the maximum expected sum of discounted future rewards at state i_{k+1}. A more detailed introduction to reinforcement learning and especially to Q-learning can be found in Bertsekas (1996) and in Watkins (1989). Glorennec (1997) and Horiuchi (1996) give a broader description of fuzzy Q-learning.

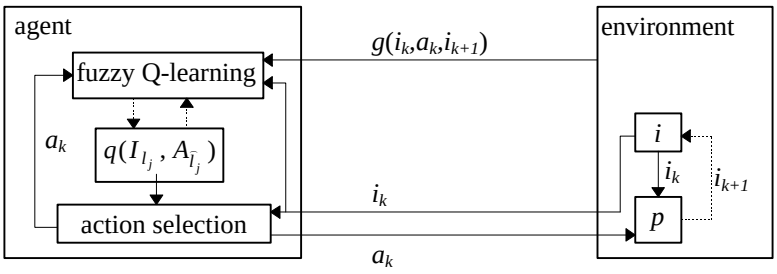


Figure 1: Framework of fuzzy Q-learning

LEARNING IN NONSTATIONARY ENVIRONMENTS

The only feedback available for an agent to detect changes in the environment are the rewards g. Thus, the basic idea of the CCR algorithm is to track the characteristics of these rewards and to learn prototypical characteristics (situations) and the appropriate strategies. Figure 2 shows the resulting two levels of the CCR approach. At the higher level, the characteristics of the feedback from the environment are evaluated. The degrees of membership m^e determined at this

level indicate the similarity between these characteristics and the characteristics of previously trained situations. If the current characteristics are highly different from those of all known situations, a new dataset, i.e. memory for the Q-values and characteristics, for a new situation is established. At the lower level, fuzzy Q-learning is executed, with the step sizes for the different datasets reduced according to the corresponding degrees of membership. The amount of memory needed to preserve the characteristics and the Q-values of the situations can be reduced by clustering similar situations. These concepts are discussed in detail in the following subsections.

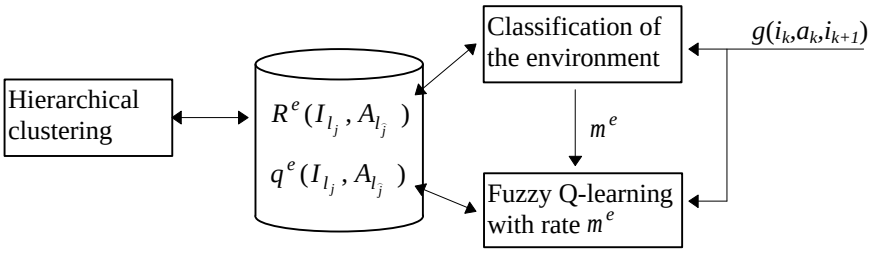


Figure 2: Architecture of the CCR algorithm

CHARACTERIZATION AND CLASSIFICATION OF THE ENVIRONMENT

The learning agent is able to observe the state transitions and the rewards of the environment. The characteristics of these observations can be trained in parallel with the Q-values and serve for the future classification of the environment. In the following it is assumed that the environment changes slowly enough so that the agent can learn characteristics of the environment before a significant change occurs. This assumption, which on first sight might be considered as a strong drawback of the proposed method, turns out to be basic for any learning-method. Even a human being can not learn an adequate behavior unless there is some stability in the reaction of its environment. A changing environment can only be distinguished from a chaotic one if its changes are sufficiently slow. As mentioned before, a dataset for a new situation is established whenever the characteristics of the current environment do not fit to any previously trained situation. Thus, the learning agent finally has to store a set of (characteristics, Q-value)-pairs. The set of all trained situations will be denoted by E, a special situation by e.

In the following, a situation e is characterized by the expectation of the rewards returned by the environment for performing action A_j in state I_j in this situation:

$$R^e(I_{l_j}, A_{\bar{l}_j}) = \mathrm{E}\left\{ g(I_{l_j}, A_{\bar{l}_j}, i_{k+1}) \right\} \tag{5}$$

These expectations can be learned by setting for each situation $e \in E$ on the experience of reward $g(i_k,a_k,i_{k+1})$ for action a_k in state i_k:

$$R^e_{k+1}(I_{l_j}, A_{\bar{l}_j}) = R^e_k(I_{l_j}, A_{\bar{l}_j}) + h^e_k m^e_k m_j \left(g(i_k, a_k, i_{k+1}) - R^e_k(I_{l_j}, A_{\bar{l}_j}) \right), \quad j = 1,\ldots,m \tag{6}$$

where m^e denotes the membership of the current environment to e, as defined below, m_j is as defined above and h is a diminishing step size.

Given the expected rewards R^e one can define the distance of the current environment to a trained situation $e \in E$ as follows:

$$d^e_k = \left| \sum_{k=0}^{\infty} l^k \left(g(i_{k-k}, a_{k-k}, i_{k-k+1}) - R^e_{k-k}(i_{k-k}, a_{k-k}) \right) \right|, \quad 0 \ll l < 1. \tag{7}$$

This distance will significantly increase if the expectation of the distribution from which g is taken changes. As mentioned before, it can be useful to consider additional characteristics of the environment like the variance of the rewards or the transition probabilities.

Based on the distances d^e one can eventually define the degrees of membership m^e of the current environment to the learned situations e:

$$m_k^e = \frac{1}{\sum_{f \in E} \left(\frac{d_k^e}{d_k^f} \right)^{\frac{2}{m-1}}} \tag{8}$$

The choice of the fuzzyfier *m* depends on the problem to solve: In a continuously changing environment, it makes sense to associate the current environment partially with several learned situations, i.e. to choose real fuzzy degrees of membership (e.g. *m*=2). On the other hand, the memberships should be crisp if the environment switches between a discrete set of individual characteristics ($m \to 1$).

DETECTION OF CHANGES IN THE ENVIRONMENT

As mentioned before, a dataset for a new situation e' is established whenever the distances d^e to all known situations $e \in E$ exceed certain thresholds dm^e. Since, due to the stochastic nature of the environment, the distances d^e are different from zero even in stationary environments, individual thresholds dm^e have to be trained for each situation *e*. This training is done by fitting $dm^{e'}$ to the maximum observed distance in the initial phase after the creation of new situations e':

$$dm_{k+1}^{e'} = \max\left(dm_k^{e'}, \min\left(d_{k+1}^{e'}, dm_k^{e'} \cdot sens(k_k^{e'}) \right) \right). \tag{9}$$

The function *sens* defines the elasticity of the upper bound $dm^{e'}$, which tends to 1 with increasing $k^{e'}$:

$$sens(k) = 1 + S_0 e^{-k/s}, \tag{10}$$

where the initial elasiticity is $S_0 + 1$ and the rate of decay is given by s .

$k^{e'}$ is increased after each application of (9) by the environmental degree of membership:

$$k_{k+1}^{e'} = k_k^{e'} + m_k^{e'}. \tag{11}$$

The idea of (9) is to adapt $dm^{e'}$ even to large increases of $d^{e'}$ at the beginning after the creation of e' and to fix it afterwards. This again presupposes that the environment does not change to fast, such that a threshold can be trained before there is a significant change.

The Q-values of a new situation e' are initialized according to the similarity of the current environment to the previously trained situations $e \in E$:

$$q_{k+1}^{e'}(I_{l_j}, A_{\bar{l}_j}) = \sum_{e \in E} m_{k+1}^e q_{k+1}^e (I_{l_j}, A_{\bar{l}_j}), \qquad j = 1, \ldots, m \tag{12}$$

Thus, initial knowledge about the new situation is derived from similar known situations, and the agent need not start learning from the beginning, i.e. from a random strategy. Nevertheless, in order to guarantee optimal behavior, one has to explore each state-action pair infinitely often in the new environment. However, if an exploration strategy is used with a randomness depending e.g. on a temperature *T* (e.g. a Boltzmann-distribution (Watkins 1989)), the initial value of *T* can be decreased according to the similarity of the new situation to a known one.

FUZZY Q-LEARNING

The training of the Q-values is done in parallel with the training of the characteristics described in the preceding subsections. On an observation $g(i_k, a_k, i_{k+1})$, the Q-values of all situations $e \in E$ are adapted according to the degrees of membership of the current real environment to the situations (cf. (4)):

$$q_{k+1}^e(I_{l_j}, A_{\bar{l}_j}) = q_k^e(I_{l_j}, A_{\bar{l}_j}) + h_k^e m_j m_k^e (g(i_k, a_k, i_{k+1}) + aV^e(i_{k+1}) - q_k^e(I_{l_j}, A_{\bar{l}_j})), \qquad j = 1, \ldots, m. \tag{13}$$

Thus, fuzzy Q-learning is performed with respect to the different situations. The generalization capability of fuzzy Q-learning allows to keep the number of explicitly distinguished situations small even in continuously changing environments.

The Q-values to be used in (2) and (3) for the exploration of the learning agent are also averaged according to the degrees of membership:

$$q_k(I_{l_j}, A_{\bar{l}_j}) = \sum_{e \in E} m_k^e q_k^e (I_{l_j}, A_{\bar{l}_j}), \qquad j = 1, \ldots, m. \tag{14}$$

Thus, the learning agent is enabled to generalize the learned Q-values and resulting strategies.

AGGREGATION OF REDUNDANT KNOWLEDGE

During learning, many similar situations can be established by the CCR-algorithm if the environment changes continuously or in small steps. If the same or a similar strategy is valid for different situations, it is not necessary to preserve detailed knowledge about each situation. Instead, it is sufficient to preserve prototypical situations.
The similarity of the current environment to known situations is determined on the basis of the characteristics R^e (cf. (7)). Thus, the criterion for the clustering of learned situations is given by the distances of their characteristics. However, situations with similar characteristics may not by aggregated into one prototype situation unless a similar or the same strategy is optimal in all these situations. This can be seen from the Q-values q^e. The latter constraint can be integrated in a hierarchical clustering. The idea is to prohibit the aggregation of clusters if the loss of information due to this aggregation exceeds a given threshold.

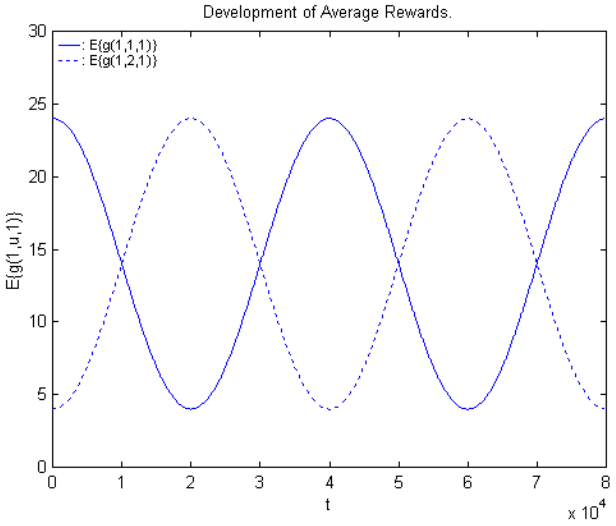


Figure 3: Development of the average rewards.

On the aggregation of situations $\{e_{i_1},\ldots,e_{i_n}\} \subseteq E$ to new prototype situations $\{e'_1,\ldots,e'_m\}$, the Q-values and R-values of $\{e'_1,\ldots,e'_m\}$ are set to the average values of the corresponding aggregated situations from $\{e_{i_1},\ldots,e_{i_n}\}$. In the following, the set of situations generated by the aggregation is denoted by $E' = E \cup \{e',\ldots,e'_m\} \setminus \{e_{i_1},\ldots,e_{i_n}\}$. Such an aggregation causes a loss of information since the Q-values for the discarded situations $\{e_{i_1},\ldots,e_{i_n}\}$ have to be interpolated from values of E' on future encounters of these situations. Small changes in the Q-values, not resulting in a change of the strategy, can be accepted unless further training of the Q-values, i.e. the Q-values of the prototype situations, is to be performed. Changes of the strategy can usually not be accepted.
The clustering algorithm is illustrated in the figures 3 and 4. It is assumed that there exists merely one crisp state, and the agent may in each step choose between two crisp actions. The agent receives rewards according to uniform probability distributions of with 8, whose expectations are shown in figure 3. The expectation of the rewards for action 1 are shown by the solid line, whereas the dotted line gives the expectation of the rewards for action 2. Obviously, the agent has to change its strategy repeatedly in time. In this changing environment, the learning agent distinguished five situations. The average rewards that the agent expects in these situations and the learned Q-values are shown in figure 4. The situations are labeled with their Q-values: [q(1,1),q(1,2)]. The position of the situations is determined by the expectation of the rewards for performing action 1 or 2 in the single state 1: [R^e(1,1), R^e(1,2)]. It can be seen, that in one situation (marked by a circle) equal rewards are expected, whereas the times that higher rewards are expected for one of the two actions are described by two situations respectively (marked by ‘x’ and squares). The whole scenario can be described by distinguishing only three prototypical situations, as is also shown in figure 4. The prototypes are marked by ‘*’, ‘+’ and the diamond. Obviously, the Q-values of the original situations will change when being interpolated from the prototypes, but this loss of information can be accepted since the strategy does not change due to it. For example, the interpolated Q-value for action 2 will still be higher than that for action 1 in one of the situations marked by ‘x’.
The same criterion that is used for the aggregation of environments can be used to determine the loss of information caused by the deletion of situations. For example, the situation describing equal expected rewards in figure 4 (marked by

‘*’) could be removed, since its Q-values could also be gained by an interpolation of the remaining (two) prototypical

The clustering and deletion of situations and the corresponding strategies can be compared with the fading and mingling of experiences and knowledge of human beings. A human being also needs to learn to distinguish (perhaps similar) situations in which different strategies are necessary, whereas situations requiring the same behavior may be mixed up.
As mentioned before, one goal of the clustering described in this subsection is to reduce the amount of memory required by the agent. In addition, the clustering allows to compensate the `false' establishing of new situations. The term `false' means, that the agent establishes a new situation although the environment does not change, which can happen because of the stochastic nature of the method described above. The Q-values and R-values of a falsely established situation will converge to values of an existing situation, such that these two situations will eventually be substituted by a single prototype without loss of information.

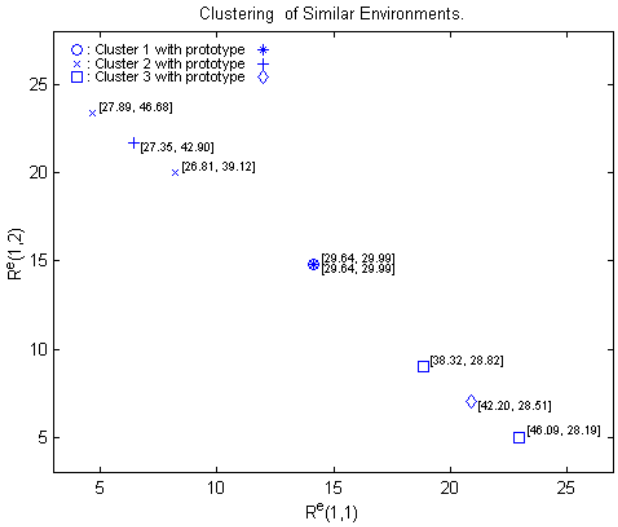


Figure 4: Clustering of similar environments.

APPLICATION TO TRAFFIC SIGNAL CONTROL

The CCR-algorithm was applied to some problems in traffic signal control. The optimization of traffic signals by reinforcement learning was first proposed by Mikami and Kakazu (1994). While Mikami and Kakazu trained a fixed pattern for the traffic signals, the idea of the following examples is to train rules that allow an adaption of the traffic signals to varying traffic flows. The following subsection will give a short description of the model of a single intersection. More complex scenarios with multiple intersections can be modeled as multi-agent systems (Ishibuchi 1997, Sandholm 1995, Tan 1993), with each agent having the proposed structure.
The remaining subsection will finally give an example of the application of the CCR-algorithm to a changing environment.

MODEL OF A SINGLE INTERSECTION

An intersection in a traffic network can be modeled as an agent that has to maximize the number of cars passing this intersection by an optimal control of the traffic signals. The state of the intersection consists of:

- Sensory information: It is assumed that the traffic density (number of cars per length unit) is measured at a finite set of points in the incoming and outgoing roads of the intersection.
- Current state of traffic signals.
- Time since last change of state of traffic signals (needed in order to prevent permanent switching).

It is assumed that the agent may choose the state of the signals at discrete points in time. The intersection gets positive rewards for the number of cars that passed the intersection due to the state of the traffic signals. Additionally, it gets negative rewards according to the number of cars waiting in front of red traffic signals. Thus, the goal is to find a strategy such that the expected discounted sum of future rewards is maximized (cf. (1)).

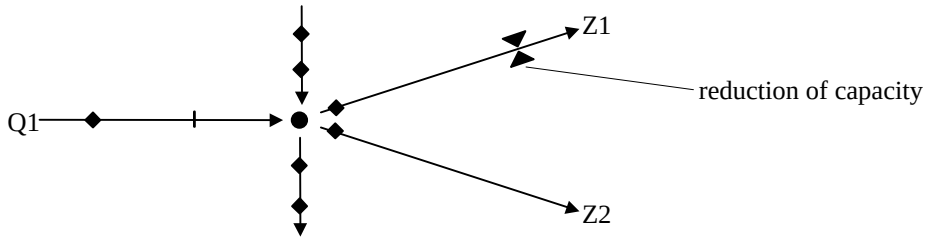


Figure 5: Simple scenario with changing OD-relation. The diamonds indicate the positions of sensors.

CHANGING OD-RELATION DURING LEARNING

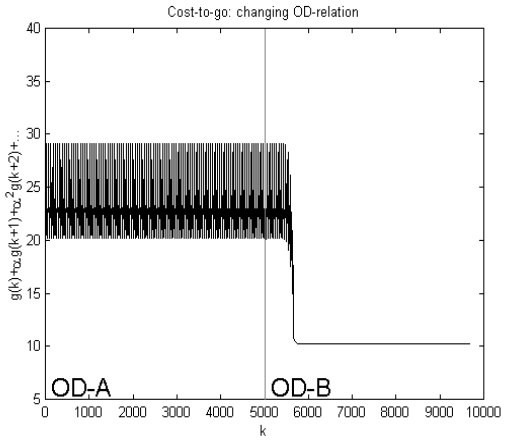
Figure 5 shows a simple scenario, in which cars arriving from Q1 may leave the intersection in two directions, Z1 and Z2. It is assumed that the fractions of cars with targets Z1 and Z2 change repeatedly from the origin-destination-relation (OD-relation) OD-A to OD-B, with OD-A and OD-B given in Table 1. It is further assumed that the capacity of the road with target Z1 is reduced, e.g. by an accident or by road repairs. This reduction does not influence the intersection if the cars have destinations according to OD-A, because few cars leave in this direction, whereas a congestion is formed if the distribution of the destinations is given by OD-B.

OD-A	**Z1**	**Z2**
Q1	0.1	0.9

OD-B	**Z1**	**Z2**
Q1	0.9	0.1

Table 1: OD-relations for scenario of figure 5.

In figure 6, the performance, i.e. the discounted sum of rewards g, of an intersection trained with ordinary Q-learning is shown. Obviously the performance is strongly reduced after the change of the OD-relation at t=5000, since the strategy learned by the agent is not suitable for OD-B. In fact, it could be observed in the experiments that after the change of the OD-relation the intersection ceases switching and preserves the horizontal connection all the time. The delay between the change in the OD-relation and the decrease of the discounted rewards in figure 6 is the time it takes for the cars with the new destinations entering at Q1 to reach the intersection, and the congestion in front of Z1 to propagate backwards into the intersection. Indeed, due to these effects, the environment cannot be considered to be purely discretely changing, but it drifts continuously for some time after the change of the inflow at Q1 and remains stable afterwards. The chattering in the performance of the intersection that can be observed in the left part of figure 6 results from the fact that it is not possible to achieve the same discounted sum of rewards in each state of the intersection.
Figure 7 shows the application of the CCR-approach to the scenario of figure 5 with repeatedly changing OD-relations. The agent is able to distinguish the two situations and to learn an individual and adequate behavior for each situation.



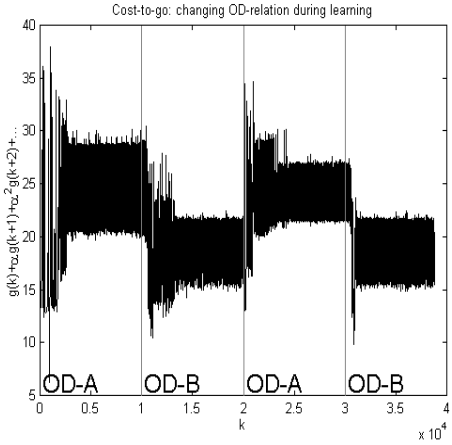


Figure 6: Discounted rewards with ordinary fuzzy Q-learning.

Figure 7: Discounted rewards with CCR-approach.

CONCLUSIONS

In this paper a new algorithm, *characterization and clustering of reinforcement signals (CCR)*, was proposed, extending fuzzy Q-learning to nonstationary environments by the implicit expansion of the state-space by characteristics of the environment. Methods for the characterization of the environment and a criterion for the automatic generation of new situations, i.e. new implicit fuzzy sets, where suggested. Initial strategies for new situations are deduced according to the similarity to known situations. Finally, a clustering algorithm for the reduction of the amount of information needed by the agent to behave properly in the nonstationary environment was proposed. The soundness of the methods was shown by an example from traffic signal optimization.
In contrast to related work done on reinforcement learning in nonstationary environments by Kaelbling (1998) or by Chrisman (1992), the method proposed in this article is a black-box approach, i.e. no restrictions are made on the dynamics of the environmen, i.e. the hidden states, and no a priori knowledge about the dynamics is needed.

REFERENCES

Bertsekas, D.P.; Tsitsiklis, J.N., 1996, ”Neuro-Dynamic Programming”, Athena Scientific.

Chrisman, L., 1992, ”Reinforcement Learning with Perceptual Aliasing: The Perceptual Distinctions Approach”, Proceedings of the Tenth National Conference on Artificial Intelligence, San Jose/California, USA, pp. 183-188.

Glorennec, P.Y.; Jouffe, L., 1997, ”Fuzzy Q-Learning”, Proceedings of the Sixth IEEE International Conference on Fuzzy Systems, pp. 659-662.

Horiuchi, T.; Fujino, A.; Katai, O.; Sawaragi, T., 1996, ”Fuzzy Interpolation-Based Q-Learning with Continuous States and Actions”, Proceedings of the Fifth IEEE International Conference on Fuzzy Systems, pp. 594-600.

Ishibuchi, H.; Nakashima, T.; Miyamoto, H.; Oh, C., 1997, ”Fuzzy Q-Learning for a Multi-Player Non-Cooperative Repeated Game”, Proceedings of the Sixth IEEE International Conference on Fuzzy Systems, pp. 1573-1579.

Kaelbling, L.P.; Littman, M.L.; Cassandra, A.R., 1998, ”Planning and Acting in Partially Observable Stochastic

Lam, W.; Mukhopadhyay, S., 1996, ”A Two-Level Approach to Learn in Nonstationary Environments”, Advances in Artificial Intelligence, Proceedings of the Eleventh Biennial Conference of the Canadian Society for Computational Studies of Intelligence, pp. 271-283.

Mikami, S.; Kakazu, Y., 1994, ”Genetic reinforcement learning for cooperative traffic signal control”, Proceedings of the First IEEE Conference on Evolutionary Computation, pp. 223-228.

Narendra, K.S.; Thathachar, M.A.L., 1989, ”Learning Automata - An Introduction”, Prentice Hall, Englewood Cliffs, New Jersey.

Sandholm, T.W.; Crites, R.H., 1995, ”Multiagent reinforcement learning in the Iterated Prisoner’s Dilemma”, Biosystems Journal, Vol. 37, pp. 147-166.

Schmidhuber, J., 1996, ”A general method for multi-agent reinforcement learning in unrestricted environments”, Adaption, Coevolution and Learning in Multiagent Systems. Papers from the AAAI Symposium (TR SS-96-01), AAAI-Press, Menlo Park, CA/USA, pp. 84-87.

Tan, M., 1993, ”Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents”, Proceedings of the Tenth International Conference on Machine Learning, pp. 330-337.

Watkins, C.J., 1989, ”Learning from delayed rewards”, Doctoral thesis, Cambridge University, Cambridge, England.