Transfer learning based on forbidden rule set in actor-critic method

Article *in* International journal of innovative computing, information & control: IJICIC · May 2011

CITATIONS
7

READS
121

6 authors, including:

Toshiaki Takano
Ritsumeikan University
15 PUBLICATIONS **118** CITATIONS
SEE PROFILE

Haruhiko Takase
Mie University
108 PUBLICATIONS **399** CITATIONS
SEE PROFILE

Hiroharu Kawanaka
Mie University
186 PUBLICATIONS **747** CITATIONS
SEE PROFILE

Shinji Tsuruoka
Suzuka University of Medical Science
166 PUBLICATIONS **1,505** CITATIONS
SEE PROFILE

International Journal of Innovative
Computing, Information and Control
Volume **7**, Number **5(B)**, May **2011**

ICIC International ©2011 ISSN 1349-4198
pp. **2907–2917**

TRANSFER LEARNING BASED ON FORBIDDEN RULE SET IN ACTOR-CRITIC METHOD

Toshiaki Takano, Haruhiko Takase, Hiroharu Kawanaka
Hidehiko Kita, Terumine Hayashi and Shinji Tsuruoka

Mie University
1577, Kurima-machiya-cho, Tsu, Mie, Japan
takano@ip.elec.mie-u.ac.jp
{ takase; kawanaka; kita; hayashi; tsuruoka }@elec.mie-u.ac.jp

Received January 2010; revised July 2010

Abstract. *In this paper, we aim to accelerate learning processes in actor-critic method. We proposed the effective transfer learning method, which reduces training cycles by using information acquired from source tasks. The proposed method consists of two ideas, the method to select a policy to transfer, and the transfer method considering the characteristic of each actor-critic parameter set. The selection method aims to reduce redundant trial and error that are used in the selection phase and the training phase. We introduce the forbidden rule set, which are detected easily in the training phase, and concordance rate that measures an effectiveness of a source policy. The transfer method aims to merge a selected source policy to the target policy without negative transfers. It transfers only reliable action preferences and state values that implies preferred actions. We show the effectiveness of the proposed method by simple experiments. Agents found effective policies from the database, and finished their training with less or same episodes than the original actor-critic method.*

Keywords: Reinforcement learning, Actor-critic method, Transfer learning

1. **Introduction.** Reinforcement learning [1] is widely used for optimization problems, for example, object manipulation problems, path search problems. Agents acquire a policy which accomplishes the target task autonomously. Many researchers try to improve reinforcement learning algorithms: reinforcement learning in noisy environment [2], cooperative reinforcement learning for multi agent [3], and so on.

Acceleration of learning processes is one of important issues in reinforcement learning [1, 4]. In reinforcement learning, agents has no information "How to solve a target task" at the beginning of their training, they should get the information from the environment by trial and error. It requires long learning processes to get enough information. Though many researchers try to accelerate learning processes [5, 6, 7], however, there are not decisive methods.

Transfer learning [8] is one of effective methods to accelerate learning processes in some machine learning algorithms. It is based on the ideas that knowledge to solve source tasks accelerate a learning process of a target task. In reinforcement leaerning, knowledge for a source task is called as a source policy. Agents acquire various source policies by training many source tasks one by one, extracts effective information from one or more source policies, and applies the information to the target task. The agent does not need to learn from scratch.

Important processes in transfer learnings for reinforcement learning are selection of effective source policies and training based on the selected policies. Mistakes in these processes make the transfer learning ineffective. Ineffective selected policies would slow

learning processes to fix them. Inappropriate transfer would disturb learning processes with inappropriate information. Thus, it is important to discuss selection method for effective source policies and appropriate transfer method. Evaluation of the effectiveness for a source policy is an important issue to discuss them.

Effectiveness of a source policy is measured by similarity between the source policy and the optimal target policy. However, the optimal policy is unknown before training. Thus, agent needs to foresee the optimal policy for evaluating effectiveness. Many researchers have tried to foresee it and measure a similarity of policies [9, 10]. Their transfer method are consist of two separated phases: selection and training. In the selection phase, agents foresee the optimal target policy and measure effectiveness for each source policy. In the training phase, agents learn the target task by using the selected policy. Since both phase independently require trial and error to get a target policy, they would waste learning process caused by redundant trial and error.

This paper aims to reduce training cycles of actor-critic method by a transfer learning. We discuss an effective transfer learning method from two viewpoints. First viewpoint is for the selection method. To reduce redundant trial and error, we try to fuse the selection phase and the training phase. Then, agents aware effective policies even in a training phase and expedite their learning process at any time. Second is the training method which is based on characteristic of each actor-critic parameter. It aims to reduce bad effect of the selected policy.

2. Acceleration a Learning Process by Transfer Learning.

In this section, we simply explain actor-critic method and a framework of transfer learning.

2.1. Actor-critic method.

Tasks in reinforcement learning are defined as Markov Decision Processes (MDPs).

Definition 2.1. *MDP is a tuple $\langle S, A, T, R\rangle$. S is a finite set of states. A is a finite set of actions. T is a stochastic state transition function $T : S \times A \times S \to \Re$, which is the probability that the action a in the state s_1 will lead the state s_2. R is a stochastic reward function $R : S \times A \to \Re$.*

Actor-critic method is one of popular reinforcement learning algorithms. It finds a policy Π, that maximizes the quantity R_t,

$$R_t = \sum_{\tau} \gamma^{\tau}\, r_{t+\tau}, \tag{1}$$

for given $\langle S, A, T, R\rangle$. Here, γ is a predefined parameter, which is called as a discount rate. A policy is a mapping $S \to A$ that brings a proper action from the current state. Here, we call individual correspondence of an action $a \in A$ to a state $s \in S$ as a rule. A policy consists of many rules.

An agent for actor-critic method consists of actor and critic (see Figure 1). Actor selects action for a current state. It observes the current state from the environment, and decides an action according to action preferences P. It tends to select an action which has higher action preference among possible actions at the current state. It holds action preferences P as a table that represents a mapping $S \times A \to \Re$. Critic evaluates actions to find a prefer action. It receives a reward, and updates state values and action preferences. State values V are held in a table that represents a mapping $S \to \Re$.

The observation, the decision and the update are repeated until the agent acquires an acceptable policy.

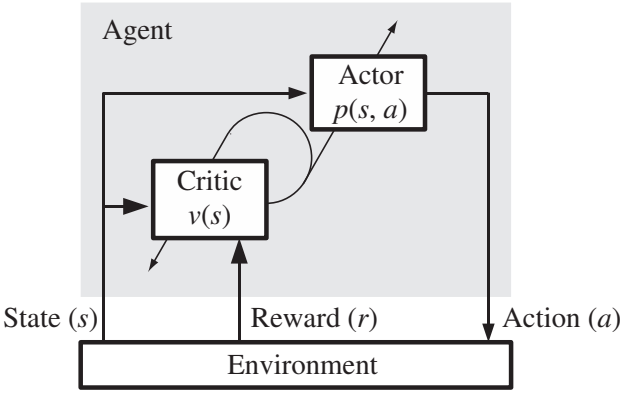

FIGURE 1. Framework of actor-critic method

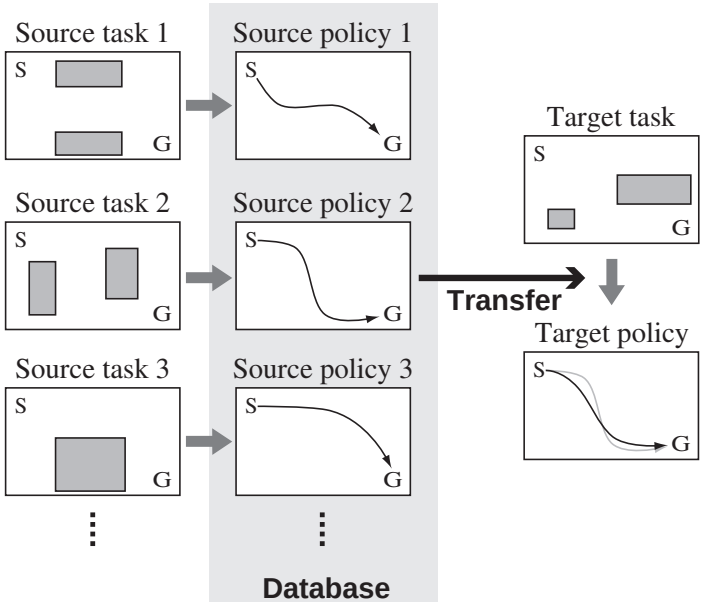

FIGURE 2. Framework of transfer learning

2.2. **Transfer learning.** In this paper, we discuss a transfer learning in actor-critic method. Figure 2 illustrates the framework of it. First, agents learn various source tasks and construct a database of policies. Second, an agent for the target task refers the database and selects a similar source policy to the optimal policy for the target task. Finally, the agent learns the target task based on the selected policy. Since the selected policy would contain effective information for the target task, the learning process of the target task would be accelerated.

We define a domain that decides acceptable source tasks. Transfer learning can transfer a policy for a source task only in the same domain to the target task.

Definition 2.2. *Domain D is a tuple $\langle S, A \rangle$. Task Ω is a tuple $\langle D, T, R \rangle$.*

The method accepts source tasks that have a same size of the state value table and the action preference table with ones for the target task. The domain is defined by many researchers independently. For example, Fernández defined a domain as a tuple $\langle S, A, T \rangle$ [10]. We intend the Definition 2.2 to keep wide application of the proposed method.

3. **Proposal.** In this section, we propose a transfer learning method in actor-critic method.

3.1. **Framework of the proposed method.** Generally, transfer learning methods in reinforcement learning consist of three phases: constructing a database that holds many policies for various source tasks, selection of an effective policy from the database and training the target task based on the selected policy.

We fuse two phases: selection and training. It aims to reduce a computational effort for these phases. Since agents have no information related to the target task before their training, they use trial and error in the selection phase to find an effective policy. Necessarily, they also use trial and error in the training phase. We aim to reduce a computational effort by reduce the duplicated trial and error. Our method consists of only two phases: constructing a database and training the target task. In the training phase, the agent trains the target task and collects clues to find effective policies, simultaneously. In the training, an agent repeats four steps:

1. The agent observes a state from the environment, decides an action and receives a reward;
2. It finds clues from the state, the action and the reward;
3. It investigates the database for an effective policy with the clues;
4. If it finds an effective policy, it transfers the policy to its parameters. Otherwise, it updates its parameters based on the original actor-critic method.

The agent accelerates its training process by transferring effective policies. Even if there are no effective policies in the database, it finishes its training without any transfers, and require a little additional computational efforts. It means that our method reduces the redundant trial and error.

3.2. **Clues to find effective policies.** The effectiveness of a source policy should be measured by the similarity between the source policy and the optimal policy for the target task. The most effective source policy is an identical policy to the optimal policy. The effectiveness is lost by increasing conflict rules, which conflict to the optimal one. Since agents get an optimal policy at the end of its training phase, they should guess it before/during their training phase. They need suitable clues to guess in the early stage of the training phase.

We introduce a forbidden rule set, which is a set of rules that cause immediate failure of a task. For each state, actions are classified into three types: preferred actions, forbidden actions and others. A preferred rule is a correspondence of a preferred action to the state. A forbidden rule is a correspondence of a forbidden action to the state. Agreement between a forbidden rule set for a source task and one for the target task would imply that preferred rules for the source task agree with one for the target task.

A forbidden rule set is a preferred clue for transfer learning because of two reasons.

First, forbidden rules are monotonically increase during the training phase. Since state values and action preferences are optimized by actor-critic method, they are changed greatly during the training phase. Hence, agents cannot find any suitable clues from them in an early stage of the training phase. Forbidden rules increase monotonically during the training phase, since forbidden actions cause failures regardless of the progress of the training. Consequently, agents can regard a forbidden rule set as a stable clue to measure the effectiveness.

Second, forbidden rule sets are easy to compute. A forbidden rule is a correspondence of a forbidden action to a state. Here, a forbidden action causes a failure of the task.

Agents can detect forbidden actions by a failure of the task during the training phase. They do not need any extra computational effort to the original actor-critic method.

We also discuss the computational effort to prepare the database. Agents evaluate the effectiveness of each policy in the database and select the most effective policy to transfer. Since agents use forbidden rule sets as clues to find an effective policy, the database should hold a policy and forbidden rule sets for each source task. As discussed above, forbidden rules are easily collected during the training phase for each source task. If there is no record of a learning process, agents can infer forbidden rules from the state values. Anyway, they need a little additional computational effort to build the database.

The effectiveness of a policy is measured by the similarity between a forbidden rule set in the database and one for the target task. Here, we introduce the concordance rate.

Definition 3.1. *The state s is an equivalent state, if all source forbidden rules related to the state s are agreed with ones for the target task. The concordance rate of the source forbidden rule set is a rate of equivalent states against all states.*

High concordance rates of a source forbidden rule set mean that the corresponding policy is effective for the target task.

Since the complete forbidden rule set for the target task is unknown during the training phase, agents compute the concordance rate based on an incomplete forbidden rule set, which is found by the instant. They select the item that has the highest concordance rate from the database, if the concordance rate is greater than the given transfer threshold θ. Here, a high threshold brings precise similarity and less transfer, and a low threshold brings opposite.

3.3. **Knowledge transfer.** Agents should avoid negative transfers, which degrade the current learning process. Negative transfers may be caused by incomplete forbidden rule sets or inappropriate transfer methods. For example, the most simple transfer method, which replaces the policy with the selected source policy, would cause a negative transfer. It transfers all rules regardless of harmful rules, which are selected according to an incomplete target forbidden rule set. Agents should transfer a policy with avoiding these problems.

We discuss a method that transfers action preferences and state values instead of a policy in the form of the set of rules. Since function of each parameter is different, they should be transferred in consideration of their characteristics.

Action preferences should be transferred carefully, since they are directly used to decide agent's action. Only reliable action preferences should be transferred. Rules that related to an equivalent state would be reliable, since all forbidden rules are agreed. The agent merges reliable rules into its action preferences by Equation (2),

$$p_t(s,a) \leftarrow (1-\zeta)p_t(s,a) + \zeta p_s(s,a), \quad \forall s \in \text{equivalent states}, \quad \forall a \in A. \tag{2}$$

Here, subscript t and s mean target and source, respectively. Transfer efficiency ζ is a fixed parameter that controls effects of the transferred action preferences. To prevent negative transfer, the transfer efficiency is defined as $0 < \zeta < 1$.

State values can be transferred aggressively. State values have less impact for the negative transfer than action preferences, since they affect agent's decision indirectly. Agents transfer only reliable action preferences, which are selected according to forbidden rules. It implies that reliable action preferences would not contain information related to preferred actions. To compensate it, preferred actions are transferred with state values. Agents transfer only positive state values, because agents tend to move to states which

```
φ → L
foreach (Ω in Ω_s) {
    initialize parameters P and V.
    φ → forbidden rule set F.
    while (agent does not satisfy termination conditions) {
        observe state s.
        decide action a.
        receive reward r.
        if (a is a forbidden action) {
            add (s, a) into F.
        }
        update P and V by actor-critic method.
    }
    add (P, V, F) into the database L.
}
```

FIGURE 3. Pseudo code to construct a database

have higher state values. They merge state values into its state values by Equation (3),

$$v_t(s) \leftarrow (1-\eta)v_t(s) + \eta v_s(s), \quad \forall s \in \{s | v_s(s) > 0, s \in S\}. \tag{3}$$

Here, transfer efficiency η is a fixed parameter that controls effects of transferred state values. As well as transfer efficiency ζ, η is defined as $0 < \eta < 1$.

3.4. **Proposed algorithm.** In this section, we show the whole procedure of the proposed method.

In the constructing a database phase, an agent learns a source task by actor-critic method, collects forbidden rules, and adds an item into the database. Here, the item means the optimal policy and the forbidden rule set for the source task. It repeats the procedure for various source tasks. Figure 3 shows pseudo code of this phase. We get the database (policy library) L from a set of source tasks Ω_s.

In the training phase, an agent learns the target task Ω_t. It searches a policy to transfer, every time it receives a reward. It transfers the policy, if the policy is different from the last selected policy. Figure 4 shows pseudo code of this phase. Agents get the optimal policy from the database L and the target task Ω_t. Here, the optimal policy is represented as the final action preferences P.

4. **Experiments.** In this section, we perform simple experiments to show the effectiveness of the proposed method. There are no previous methods that has the same domain definition. Thus, we compare our method with original actor-critic method and simple transfer methods.

4.1. **Experimental setup.** We use simple maze tasks for our experiments. Each maze consists of 7×7 cells. Each cell is a coordinate or a pit. An agent moves from the start cell to the goal cell through only coordinates. The agent moves 4-way one-by-one, and decides its action by sensing its location. It repeats observation, decision and action, every time it moves one cell. Here, the domain D is defined with $S = \{S_1, S_2, \ldots, S_{49}\}$ and $A = \{\text{up}, \text{down}, \text{left}, \text{right}\}$. State labels are arranged in a row major way from the left upper corner to the right bottom corner. The state S_9 is the start cell and S_{41} is the goal cell for all tasks. Rewards are defined as follows: –50 for actions to get out of coordinates,

```
initialize parameters P and V.
φ → forbidden rule set F
( ) → the latest transferred item (P_p, V_p, F_p).
while (agent does not satisfy termination conditions) {
    observe state s.
    decide action a.
    receive reward r.
    if (a is a forbidden action) {
        add (s, a) into F.
    }
    ( ) → the most effective item (P_e, V_e, F_e).
    0 → the highest concordance rate C_e.
    foreach ((P_d, V_d, F_d) in database L) {
        concordance rate for F_d to F → C.
        if (C > C_e) {
            (P_d, V_d, F_d) → (P_e, V_e, F_e).
            C → C_e.
        }
    }
    if (C_e > θ && (P_e, V_e, F_e) != (P_p, V_p, F_p)) {
        merge P_e to P according to Equation (2).
        merge V_e to V according to Equation (3).
        (P_e, V_e, F_e) → (P_p, V_p, F_p).
    } else {
        update P and V by actor-critic method.
    }
}
```

FIGURE 4. Pseudo code to learn the target task

FIGURE 5. Mazes for target tasks

100 for actions to reach the goal, and –25 for actions every 100th move. State transition T is defined as follows. For all moves that are same to agent's actions, transition rate is 0.9. Agents turn right against their actions by the transition rate 0.05, turn left in the same manner. They never move to opposite to agent's actions and remain stationary.

We prepare three mazes for target tasks (see Figure 5) and 24 mazes for source tasks (see Figure 6). In these figures, white cells are coordinates, and black cells are pits. First, we prepare a database by training an agent for each source task. The database is commonly used for following experiments.

An agent finishes its learning process, when it reaches to the goal cell for ten episodes in a row. Each episode is a subsequence of the learning process while the agent moves from the start to a pit or the goal. Parameters of actor-critic method are as follows: discount rate $\gamma = 0.95$, learning rate $\alpha = 0.05$, step size parameter $\beta = 0.05$. The agent decides

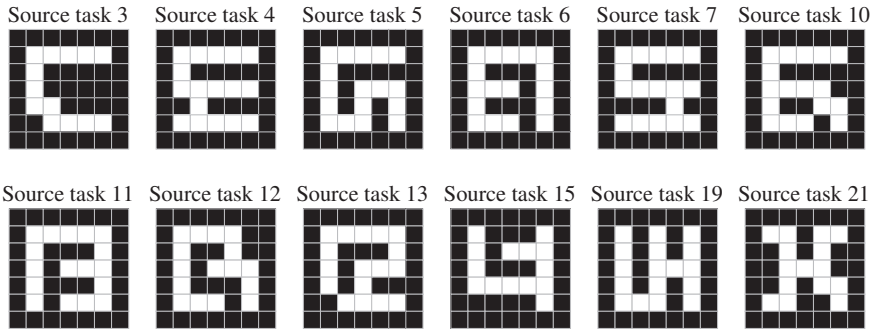

FIGURE 6. Sample mazes for source tasks

TABLE 1. Number of episodes for each transfer method

(a) Task A

V\P	Proposed Transfer	Simple Transfer	No Transfer
Proposed Transfer	177.0	NA	277.4
Simple Transfer	811.8	NA	444.9
No Transfer	NA	NA	284.5

(b) Task B

V\P	Proposed Transfer	Simple Transfer	No Transfer
Proposed Transfer	186.0	NA	213.9
Simple Transfer	NA	215.5	314.2
No Transfer	271.6	NA	261.1

(c) Task C

V\P	Proposed Transfer	Simple Transfer	No Transfer
Proposed Transfer	210.1	NA	307.1
Simple Transfer	NA	NA	312.0
No Transfer	138.2	NA	283.3

its action by soft-max method during its learning. The transfer threshold θ is 0.2. The fixed transfer efficiency ζ and η is 0.5, 0.05, respectively.

4.2. **Acceleration of learning processes.** In this section, we discuss the effect of the proposed method. Agents learn each target task by nine learning methods that are different in the transfer method. Each learning uses the plain method, simple transfer, or proposed transfer for action preferences and/or state values. Here, the plain method does not use any transfer, and the simple transfer means to transfer all parameters. We evaluate the effectiveness of each method by the number of episodes to finish a learning process. We perform 100 learning processes for each transfer method with different random sequences. Each learning process is finished when the agent acquires an acceptable policy or tries 2,000 episodes. The later is regarded as a failure of the learning.

Table 1 shows the result for the target task A, B and C, respectively. Each value represents the average number of episodes. Here, "NA" means the result with many

failures, more than 20 failures in 100 learning process. Grayed cells mean results that shows significant differences ($p < 0.05$) from the plain method (right bottom cell).

Only the proposed method, which transfers both action preferences and state values, does not bring any worse results than the plain method. Simple transfer for action preferences and/or state values causes failure of learning or negative transfer for all target tasks. Other methods bring worse results than the plain method occasionally.

These results imply the effectiveness of the proposed method. It does not degrade its learning process and would bring acceleration of the process.

4.3. **Selected policy.** In this section, we discuss an effectiveness of selected policies. We measure the effectiveness of each source policy by the rate of consistent rules with the target rules. Here, both source policy and target policy are optimal, which are found by hands.

Table 2 shows transferred policies for each target task by the proposed method ($\theta = 0.2$). ID and effectiveness mean the task ID whose policy is used, and the rate of consistent rules in the rule of selectable actions for the target task, respectively. The frequency of transfer is a frequency in one hundred learning processes for each policy. Total and last in the "Freq. of transfer" row means the total frequency and the frequency of the latest policy in each learning process, respectively. The bottom of each table shows basic statistics of effectiveness for all policies in the database.

These results show that the proposed method transfers effective policies, though there are some exceptions. For the task B, all transferred policies have the highest effectiveness in the database. For the task A and C, some ineffective policies are transferred. Agents cannot avoid some ineffective transfers, since the proposed method selects policies based on the incomplete target forbidden rules. Target forbidden rules increase with a progress of the learning. The accuracy of the effectiveness should be increased as target forbidden rules increase. The results indicate the most of the latest transferred policy (94% for the task A and 96% for the task C) have a higher effectiveness than their average. These results imply that the proposed method measures effectiveness of each policy correctly based on the concordance rate during learning processes.

5. **Conclusions.** In this paper, we reduce episodes to learn a target task by actor-critic method. We proposed the effective transfer learning method that consists of two basic idea: the selection of a policy for transfer, and the transfer method considering the characteristic of each actor-critic parameter.

The selection method finds effective policies according to the concordance rate. The concordance rate means the rate of states that are regarded as reliable. They are calculated from forbidden rule sets. An agent finds them by observing rewards during its learning.

The transfer method merges action preferences and state values of the selected policy to the current parameters. To avoid negative transfer, these parameters are transferred considering their characteristic. Since action preferences are used to decide actions, they are transferred carefully. Only reliable action preferences are transferred. They are judged by forbidden rules. State values are used to adjust action preferences. Since they have less influence to the decision than action preferences, they are transferred aggressively. Positive state values are transferred to lead agents to preferred actions. These actions would be preferred actions for the target task.

The effectiveness of the proposed method is shown by some simple experiments.

TABLE 2. Transferred Policies

(a) Task A

ID	Effectiveness	Freq. of transfer	
		Total	Latest
4	0.65	12	7
5	0.76	7	4
6	0.06	29	3
7	0.76	30	5
8	0.65	1	0
9	0.47	2	1
10	0.56	81	52
11	0.29	1	0
12	0.18	4	0
13	0.18	5	0
24	0.35	7	4
	Max. 0.88 Min. 0.06 Ave. 0.43		

(b) Task B

ID	Effectiveness	Freq. of transfer	
		Total	Latest
21	0.79	33	33
	Max. 0.79 Min. 0.07 Ave. 0.21		

(c) Task C

ID	Effectiveness	Freq. of transfer	
		Total	Latest
5	0.33	2	0
13	0.27	1	0
14	0.07	3	2
15	0.33	65	54
19	0.33	8	5
	Max. 0.53 Min. 0.00 Ave. 0.27		

REFERENCES

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning*, MIT Press, Cambridge, 1998.

[2] M.-R. Kolahdouz and M. J. Mahjoob, A reinforcement learning approach to dynamic object manipulation in noisy environment, *International Journal of Innovative Computing, Information and Control*, vol.6, no.4, pp.1615-1622, 2010.

[3] P. Darbyshire and D. Wang, Effects of communication in cooperative Q-learning, *International Journal of Innovative Computing, Information and Control*, vol.6, no.5, pp.2113-2126, 2010.

[4] L. P. Kaelbling, M. L. Littman and A. W. Moore, Reinforcement learning – A survey, *Journal of Artificial Intelligence Research*, vol.4, pp.237-285, 1996.
[5] M. Wiering and J. Schmidhuber, Fast online Q(λ), *Machine Learning*, vol.33, pp.105-115, 1998.
[6] A. P. de S. Braga and A. F. R. Araújo, Influence zones – A strategy to enhance reinforcement learning, *Neurocomputing*, vol.70, pp.21-34, 2006.
[7] L. Matignon, G. J. Laurent and N. le Fort-Piat, Reward function and initial values – Better choices for accelerated goal-directed, *Lecture Notes in Computer Science*, vol.4131, pp.840-849, 2006.
[8] S. J. Pan and Q. Yang, A survey on transfer learning, *Technical Report, HKUST-CS08-08*, 2008.
[9] B. Price and C. Boutilier, Accelerating reinforcement learning through implicit imitation, *Journal of Artificial Intelligence Research*, vol.19, pp.569-629, 2003.
[10] F. Fernández and M. Veloso, Probabilistic policy reuse in a reinforcement learning agent, *Proc. of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems*, pp.720-727, 2006.