An approach to Lifelong Reinforcement Learning through Multiple Environments

Fumihide Tanaka * **and Masayuki Yamamura** **

Tokyo Institute of Technology, Interdisciplinary Graduate School of Science and Engineering, Department of Computational Intelligence and Systems Science, (vun@es.dis.titech.ac.jp)

**Tokyo Institute of Technology, Interdisciplinary Graduate School of Science and Engineering, Department of Computational Intelligence and Systems Science, (my@dis.titech.ac.jp)*

Abstract. *Autonomous-robots* are expected to be useful in such unknown environments as space planets. For these robots, the learning ability is an essential function. *Reinforcement Learning* is considered to be one of the approaches to realize them. However, as the number of trials is needed to be large to learn, it is actually very difficult to use it in the practical environments. We think it is difficult eternally to solve this issue as long as its framework focuses on single-task learning like most conventional machine learning approaches. In this paper, we present a reinforcement learning approach dealing with multiple environments as multiple-tasks. We show some computer simulation results as an example which supports the effectiveness of our approach. Though they are computer simulations, we think that they give us good clues to the practical use in the real world. Then at last, we also mention an expansion of this idea which have the possibility of making a paradigm in this field. We hope this consideration can be the first step toward the researches of real autonomous-robots.

1 Introduction

Outer space, the deep sea,... There are still many unexplored worlds for humans. In such environments, it is *Autonomous-robots* that are expected to achieve much. A robot cannot know the whole of such unknown and uncertain environments; then it needs to be capable of autonomous learning.

Reinforcement Learning is a framework which is suitable for the learning of such autonomous-robots. The features of reinforcement learning are that a learner uses delayed reward as a clue, learning autonomically through the repeated trial and error, and thus deals with the uncertainty of the environments. Though it has often been said recently that the reinforcement learning framework should be applied to real-world problems, it still has not become the strong technique in solving such problems because of the huge numbers of trials. We think that it is difficult eternally to solve this issue as long as its framework focuses on the learning of single-task like most conventional machine learning approaches.

On the other hand, there have been attempts of machine learning to handle multiple-tasks recently (Caruana,1996) (Thrun,1996b). The lifelong learning framework (Thrun,1995) assumes the multitude of related tasks to learn. There, the performance of the n-th task is improved by employing knowledge gathered in the previous $n-1$ tasks. To gather knowledge, explanation-based neural network learning(EBNN) algorithm (Thrun,1996a) is used. They applied this framework in some fields including the control of robots by reinforcement learning (Thrun,1996a). They assumed a fixed environment and regarded the different policies under the environment as the n-tasks.

It is a very interesting and promising idea to apply the thought of handling multiple-tasks to the reinforcement learning framework. Reinforcement learning has been waiting for such an idea. We try to approach this **Lifelong Reinforcement Learning(LRL)** framework from another point of view. In contrast with above idea, we regard the different environments as the n tasks. For example, they

correspond to the n mazes. We assume n environments that have some relations in common, and a learner gathers knowledge through the previous $n-1$ tasks as the bias of the n-th task. Imagine the robot navigation in the mazes. For example, only the places of start and goal are fixed and other environmental factors, i.e. the places of obstacles, the size of maze,..., are determined randomly. Under such situations, if their relations were gathered properly through the $n-1$ mazes, then the n-th maze would be solved efficiently by employing them as the bias. **This corresponds to such scenario that the autonomous-robot is pre-trained in small pseudo-environments in some laboratories or a space probe before it is sent to unknown planets.**

In this paper, we propose an approach to the LRL framework using the stochastic gradient method. Section-2 presents this approach. Then its effects are shown by computer simulations in section-3. Future directions are mentioned in section-4 and they are the expansion of this idea which we are now undertaking. We think that these idea have the possibility of making a paradigm in the field of reinforcement learning. At last, section-5 is a conclusion of this paper.

2 Lifelong Reinforcement Learning

The lifelong learning framework assumes the n related tasks. When the learner faces the n-th task, he employs knowledge gathered in the previous $n-1$ tasks to improve his performance (Thrun,1995).

Here, we assume n different environments to be n tasks that are solved by the reinforcement learning approach individually. The n tasks have common relationships, and through the learning of $1 \sim n-1$ tasks, they are acquired as the bias to the next (n-th) task. So in this paper, we divide the n tasks into the following two phases.

$$\begin{array}{rcl} 1 \sim n-1 \text{ tasks} & \rightarrow & \textit{Acquiring bias phase} \\ n\text{-th task} & \rightarrow & \textit{Main learning phase} \end{array}$$

The main purpose of this learning is to accelerate the learning of *Main learning phase* by using the bias acquired in *Acquiring bias phase*.

The following presents an actual technique. 2.1 explains the reinforcement learning algorithm that is used in each task respectively. 2.2 describes how the biases are acquired through the tasks. 2.3 shows how the biases are used in *Main learning phase*.

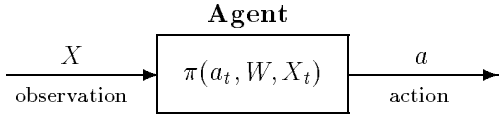
2.1 Stochastic Gradient Method

To solve each task, we use the stochastic gradient method (Kimura,1995), which is a type of memoryless reinforcement learning (Jaakkola,1994). The general algorithm is shown in Figure 1.

1. Input observation X_t.
2. Select an action a_t according to the probability $\pi(a_t, W, X_t)$.
3. Get the reward r_t from the environment.
4. Calculate $e_i(t)$ and $D_i(t)$ with all factors w_i in inside-variable W.
$$e_i(t) = \frac{\partial}{\partial w_i} ln\{\pi(a_t, W, X_t)\}$$
$$D_i(t) = e_i(t) + \gamma D_i(t-1)$$
$(0 \leq \gamma \leq 1$: discount factor)
5. Calculate $\Delta w_i(t)$ as below.
$$\Delta w_i(t) = (r_t - b) D_i(t)$$
(b : base reward)
6. Improve the policy by updating W.
$$\Delta W(t) = \{\Delta w_1(t), \Delta w_2(t), ...\Delta w_i(t)...\}$$
$$W \leftarrow W + \alpha(1-\gamma)\Delta W(t)$$
(α : learning rate)
7. Increment t and go to 1.

Figure 1: Stochastic Gradient Method

In this algorithm, the learner calculates the stochastic policy (the probability that he selects an action) when he observes environmental inputs X_t (Figure 2). To calculate this, various non-linear functions such as fuzzy-inference, neural networks, ...(and naturally, lookup-tables) can be used. After selecting an action in accordance with the policy, the learner calculate the eligibility (Williams,1992) e_i and its history D_i (eligibility trace (Singh,1994) (Singh,1996)). Then if he gets the reward r_t, the weights of non-linear function (W) are updated according to the D_i.



(W: inside-variable)

Figure 2: Stochastic policy

Kimura proved the fundamental theorem that guarantees that the eligibility trace is equal

to the gradient of expected discounted reward of POMDPs(partially observable Markov decision processes) under the fixed policy (Kimura,1995). We think that this method is desirable for reinforcement learning in unknown non-Markovian environment in the sense that it supplies an approximately rational behavior to improve the activity. (Other approaches to POMDPs: (Kaelbling,1996))

Many conventional works in reinforcement learning(*ex.* (Sutton,1988) (Watkins,1992)) were mainly aimed at MDPs(Markov decision processes). But real-world decision tasks are essentially non-Markovian. So we use this stochastic gradient method because of its expandable ability to them.

2.2 Acquiring bias

We use an artificial neural network as the representation of a non-linear function $\pi(a_t, W, X_t)$. We focus upon its weights. For example, if the neural network consists of 2 layers(*inputs*$\times 9$, *outputs*$\times 8$), then it has 72 nodes(weights). (Figure 3)

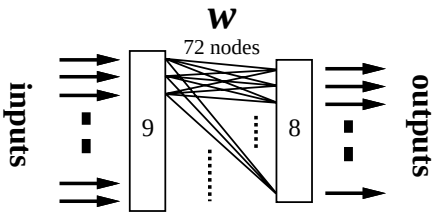


Figure 3: Neural network (2 layers)

As the bias, we pay attention to the variable range of each weight throughout the n tasks. For example, if the weight of a node varies little throughout the tasks, it can be estimated as an invariant-node (Figure 4, upper), and if it varies much, then it can be estimated as a task-dependent-node (Figure 4, lower).

In *Acquiring bias phase*, the following two biases about the converged weights are calculated on every 72 nodes.

1. The average weight → initial bias ··· ①
2. The dispersion of weights → learning bias ··· ②

Then they are built in *Main learning phase*.

2.3 Use of the bias

Each of the two biases(①,②) has its own role in *Main learning phase*.

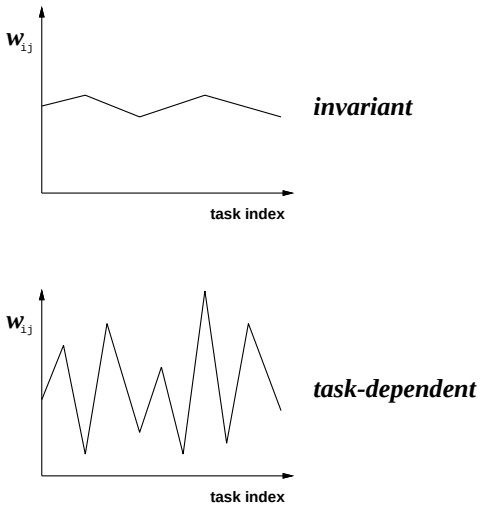


Figure 4: The variation of each weight

First, ① is used as the initial value (in each node) for the neural network of the n-th task. In reinforcement learning, the initial-randomwalk-stage is usually very expensive. So this bias is oriented to reduce the costs.

Second, ② is built in the algorithm of the stochastic gradient method as tuning parameter for the learning rate(α) of each weight. This time, the dispersions are calculated in simply, as below.

$$\beta_{ij} = \epsilon\,(1 + w_{ij}^{max} - w_{ij}^{min}) \qquad (1)$$

Here the weights were standardized($0.0 \sim 1.0$) in advance. (i, j: node index, ϵ: bias parameter) By using this bias, the updating equation of the stochastic gradient method is changed as follows.

$$W \leftarrow W + \alpha\,\beta_{ij}\,(1-\gamma)\Delta W(t) \qquad (2)$$

By using this equation, each weight is updated respectively according to the each bias β_{ij}.

3 Experiments

In this section, the experimental results that show the effect of using bias are presented. We do two kinds of experiments by changing the variety of tasks in *Main learning phase*.

3.1 Plan

First, we give the agent 9 visual-inputs (including 1 verbose input: for the improvement of generalization ability of the neural network) and 8 actual-

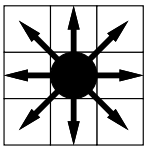


Figure 5: Agent's view and actions

directions(Figure 5). The topology of its neural network is like Figure 3.

Second, we prepare 10 mazes for the learning tasks in *Acquiring bias phase*. The size of the mazes (8×8) and the positions of start and goal are fixed, and they all have 20 obstacles of which the positions are determined randomly.

After acquiring the biases in *Acquiring bias phase*, the learner tries the mazes of *Main learning phase*. We have prepared two sets of mazes in *Main learning phase*. One is used in experiment-1 and the other is in experiment-2.

With all the mazes in *Main learning phase*, these four kinds of algorithms are tested.

1. use no bias
2. use only initial bias (①)
3. use only learning bias (②)
4. use both biases

All experiments are done 10 times and their average results are compared.

3.1.1 Experiment-1

The aim of this experiment is to find the role of each bias.

There are 5 mazes of *Main learning phase* (Figure 6) : 4 mazes of which the obstacle-mapping was predetermined, and 1 maze randomly generated. The size of the mazes and the positions of start and goal are the same as in the above mazes.

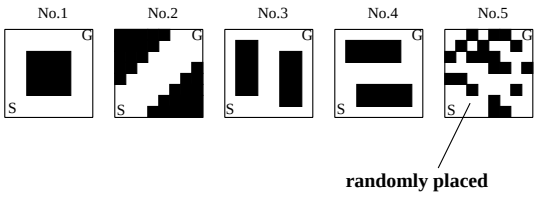


Figure 6: *Main learning phase* in experiment-1

3.1.2 Experiment-2

The aim of this experiment is to inspect the character and the extensibility of the bias.

This time, we focus on the scenario mentioned before(section-1), and the size of the maze in *Main learning phase* is changed. Using the same biases as in experiment-1, that is to say using the biases acquired in 8×8 mazes, the learner faces the maze whose size is extended to 10×10, 20×20 and 30×30. The density of obstacles is constant and the positions of them are determined randomly except the start and goal.

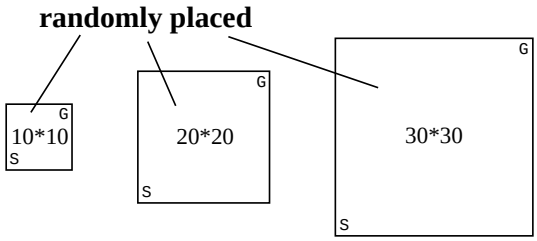


Figure 7: *Main learning phase* in experiment-2

3.2 Results

3.2.1 Experiment-1

The results of the experiment-1 are shown in Table 1. The values of the table are the average-steps until 50th goal. In every case, the biases improve the performance in the same way. Figure 8 shows the effect of the biases on the maze No.1. In every maze, the shape of the graph was almost the same.

Table 1: Results of experiment-1

maze	No.1	No.2	No.3	No.4	No.5
no bias	2808	2348	2436	2422	2623
only initial bias	1591	1230	1358	1409	1371
only learning bias	1430	1179	1378	1284	1428
both biases	**1070**	**947**	**984**	**1005**	**1034**

3.2.2 Experiment-2

The results are shown in Table 2 and Figure 9. In these experiments, the biases are effective too even though the size of the maze in *Main learning phase* is increasing. In Figure 9 (this is the results of Maze 30×30), only two methods are plotted because the methods that use no initial bias are so costly in

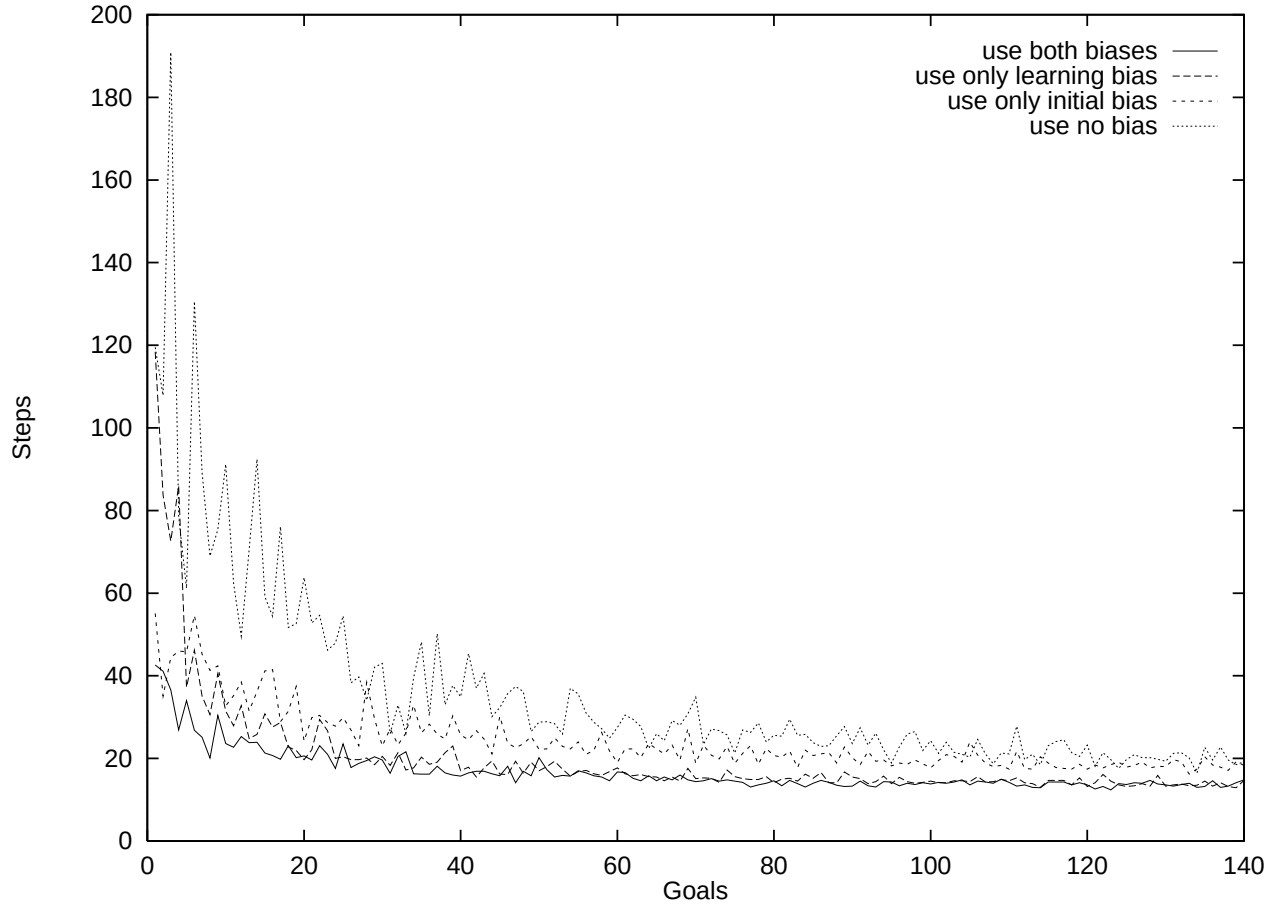


Figure 8: Transitions of the average-steps in Maze No.1

initial-randomwalk-stage of learning that the comparison between methods becomes difficult. Basically, the general tendency of the graphs in these experiments are the same as in the previous experiments(Figure 8).

Table 2: Results of experiment-2

maze	10×10	20×20	30×30
no bias	4529	12440	27890
only initial bias	1945	4568	7743
only learning bias	2292	5290	10679
both biases	**1466**	**2970**	**5220**

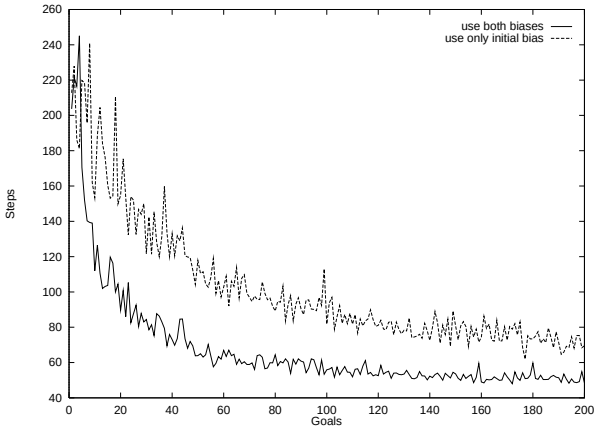


Figure 9: Maze 30×30

3.3 Discussion about the results

Both experiments show the effect of using bias. Here we discuss the results respectively.

1. In first experiments, there are some interesting results shown in Figure 8. Four lines are plotted. Among these, when initial bias is used (case-2 and 4 of section 3.1), the performances at the initial-stage is improved dramatically. When learning bias is used (case-3 and 4), the performance of the final-stage is improved in turn. The performance offset between the graph of case-1,2 and case-3,4 remains even if the experiments are continued longer. Then when both biases are used (case-4), it is shown that the two effects are combined and out of the four methods, the optimum performance is obtained.

2. Second experiments indicate the real utilities. In the real-world, the size of task is mostly unknown in contrast with the limited space of our laboratory. These experiments present the possibility of this approach to such a situation. The learner can use the biases that were obtained in smaller environments such as laboratories or space probes when he is sent to unknown environments.

These two experiments firstly aim to find out whether such biases are meaningful or not. In the real-world, the totally unknown environment is rare. If there exists some clue (such as the rough direction toward the goal), it can be used as the bias to the new task. The results of our experiments show an example of such an effect.

4 Future works

We proposed an approach of lifelong reinforcement learning in this paper. Based on this idea, we are now undertaking the expansion of it by paying more attention to the meanings of "lifelong".

The important points are summarized like below.

Table 3: Expansion of the idea

	This paper	Undertaking
life cycle (length)	short (fixed)	long (theoretically ∞)
related factor	static	dynamic

Imagine a working robot in a space planet. He works continuously to do some tasks every day. He stays working there for relative long periods (several weeks, months, years,..), and so he is an lifelong agent. As his life cycle is long, there may happen some environmental changes, so the environments has the character of dynamism.

In Table 3, the "long" life cycle is composed of many *short life cycles (slc)*. Each *slc* corresponds to the *Acquiring bias phase* in this paper and then each has its own bias respectively(Figure 10). The related factor means the relations between the tasks. In this paper, it was the geological invariance that the positions of start and goal are fixed throughout the tasks.

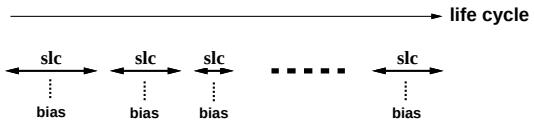


Figure 10: *short life cycles* and acquiring biases

As shown in this paper, the biases of our approach represent the features of environments that the agent lives in.

If the lifelong agent can find the transition of them (which were acquired in many *slc*) and use it well, he will be able to follow the dynamical environments. One of the example of these environments correspond to such that the positions of start and goal are moved gradually throughout the tasks.

To make the agent handle these situations, we are now investigating these topics below.

1. How does he control the length of *slc*?

2. How does he grasp the transition of the biases?

With these topics, an approach we think is like that the length of *slc* is in inverse proportion to the differential of the previous biases.

$$\frac{1}{\frac{\partial}{\partial T}\{bias\}} \cdot \eta = slc\_length \tag{3}$$

Here η is a parameter. In the lifetime of an agent, when the transition of the bias becomes large, then the environment is changing(dynamic) and the length of next *slc* should be shorten. Conversely, when the bias is almost constant, then the environment is static and it should be made longer.

In the real-world, the environments are dynamic in most cases. A navigation robot may move a goal unconsciously when he reaches there. The geographical features of a planet may be changed

gradually. **The ability to deal with the dynamic environments is essential function to the "lifelong" agent.**

5 Conclusion

We aim to apply reinforcement learning framework to real-world problems. For that purpose, we think that the "lifelong" concept is needed in its framework to reduce the huge numbers of trials.

This paper presented a reinforcement learning approach dealing with multiple-tasks. The key concepts are:

- Multiple environments are regarded as multiple-tasks.
- If some clues can be found about unknown environments, the learning performance can be improved by using the biases acquired by training in similar environments.

We did some experiments that show the effects of our approach.

1. The two biases we presented had their own role to improve the performance and by using them together, it was improved dramatically.
2. The acquired biases can be used in larger environments; this shows a possibility of applying the approach to real-world problems.

Based on this idea, we also mentioned at last the expansive views that investigated the "lifelong" concept of an agent more generally.

Though our works are tested only in the computer simulations at the moment, we think that they give us good clues to the practical environments. We hope the framework of **Lifelong Reinforcement Learning** can be a useful paradigm of real autonomous-robots.

References

R. Caruana. (1996). Algorithms and Applications for Multitask Learning. *Proc. of the 13th Int. Conf. on Machine Learning*, pp.87-95 (1996)

T. Jaakkola, S.P. Singh and M.I. Jordan. (1994). Reinforcement Learning Algorithm for Partially Observable Markov Decision Problems. *Advances in Neural Information Processing Systems*, pp.345-352 (1994)

L. Kaelbling, M. Littman and A. Moore. (1996). Reinforcement Learning: A Survey. *Journal of AI Research*, Vol.4, pp.237-285 (1996)

H. Kimura, M. Yamamura and S. Kobayashi. (1995). Reinforcement Learning by Stochastic Hill Climbing on Discounted Reward. *Proc. of the 12th Int. Conf. on Machine Learning*, pp.295-303 (1995)

S.P. Singh, T. Jaakkola and M.I. Jordan. (1994). Learning Without State-Estimation in Partially Observable Markovian Decision Processes. *Proc. of the 11th Int. Conf. on Machine Learning*, pp.284-292 (1994)

S.P. Singh and R.S. Sutton. (1996). Reinforcement Learning with Replacing Eligibility Traces. *Machine Learning*, Vol.22, pp.123-158 (1996)

R.S. Sutton. (1988). Learning to Predict by the Methods of Temporal Differences. *Machine Learning*, Vol.3, Nos.2/3, pp.9-44 (1988)

S. Thrun and T.M. Mitchell. (1995). Learning one more thing. *Proc. of the 14th Int. Joint. Conf. on Artificial Intelligence*, pp.1217-1223 (1995)

S. Thrun. (1996). Explanation-Based Neural Network Learning: A Lifelong Learning Approach. *Kluwer Academic Publishers*, (1996)

S. Thrun and J. O'Sullivan. (1996). Discovering Structure in Multiple Learning Tasks: The TC Algorithm. *Proc. of the 13th Int. Conf. on Machine Learning*, pp.489-497 (1996)

C.J.C.H. Watkins and P. Dayan. (1992). Technical note: Q-Learning. *Machine Learning*, Vol.8, No.3, pp.279-292 (1992)

R.J. Williams. (1992). Simple Statistical Gradient Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*, Vol.8, pp.229-256 (1992)