

IMPROVING REINFORCEMENT LEARNING OF AN OBSTACLE AVOIDANCE BEHAVIOR WITH FORBIDDEN SEQUENCES OF ACTIONS

C. Touzet, N. Giambiasi

DIAM - IUSPIM

Domaine universitaire de St Jérôme
F - 13397 Marseille cedex 20, France

Email: diam_ct@vmesa11.u-3mrs.fr
samira@eerie.fr

S. Sehad

LGI2P

EERIE, Parc Scientifique G. Besse
F - 30 000 Nîmes, France

Email:

Abstract

This paper is concerned with the improvement of reinforcement learning through the use of forbidden sequences of actions. A given reinforcement function can generate multiple effective behaviors. Each behavior is effective only considering the cumulative reward over time. It may not be the behavior expected by the designer. In this case, the usual solution is to modify the reinforcement function so as to introduce a new constraint related to the behavioral aspect to express. Alternatively, we propose not to modify the reinforcement function, but to add an external module containing generic forbidden sequences of actions. Experiments with the real miniature robot Khepera in a task of learning an obstacle avoidance allow to confirm the interest of this approach.

Key Words

Reinforcement learning, obstacle avoidance, sequence, self-organizing map, autonomous robotics

1. Introduction

This paper is concerned with the application of reinforcement learning to the control of a real robot that acts in a real environment. Lots of references are available for improving the performance of the learning algorithm, but few works have been done on the design of the reinforcement functions. Reinforcement functions are usually hand-tuned and emerge after lots of experiments. The reinforcement

learning algorithm task is to improve the cumulative reward over time. Despite all efforts, it happens too often that the resulting behavior maximize rewards, but does not express the expected behavior. In this case, the usual solution is to modify the reinforcement function so as to introduce a new constraint related to the behavioral aspect to express. It may also be impossible, in particular if the behavior must take into account long sequences of actions. Alternatively, we propose not to modify the reinforcement function, but to add to the learning an external module containing generic forbidden sequences of actions.

In this paper, we will only discuss a "simple" task: the learning of an obstacle avoidance behavior for the miniature robot Khepera. The reinforcement function has been given in [1]. It does not express any constraints on the covered distance. A self-organizing map implements the Q-learning. Other reinforcement learning methods, like AHC [2] may be more effective on the chosen task, but we are only looking for an illustration of the use, and of the interest, of forbidden sequences in the improvement of the matching between expected and obtained behavior.

Our general methodology is the following. First, we select a real environment, a robot and a target behavior that we want the robot to exhibit in the environment. Then, we design a reinforcement function that is the same for all experiments. Finally, we execute a number of experiments to see whether the expected behavior emerges and we analyze the effect of

forbidden sequences of actions on the robot's behavior.

In this paper, we present the results of a research aimed at improving reinforcement learning through the use of forbidden sequences of actions. In section 1, the tool and environment of our experiments is presented, in particular the miniature robot Khepera. Section 2 presents the reinforcement function used here for the learning of an obstacle avoidance behavior. In section 3, a self-organizing map implementation of Q-learning gives results that will serve us as benchmark. Results of the learning are reported in two different ways: evolution of the Q-values and performance over time. The resulting behaviors, presented in section 4, exhibit a great diversity of covered distances. Explanations are reviewed. Section 5 proposes the use of forbidden sequences of actions to force the learning of the expected behavior. In section 6, several experiments allow to demonstrate the interest of forbidden sequences of actions. Concluding remarks are given in section 7.

2. The miniature robot KHEPERA

Khepera is a miniature robot [3] having a diameter of 6 cm and a weight of 86 g (Fig. 1). Two independent wheels allow the robot to move around. The number of possible actions is reduced to four hundreds. Eight infra-red sensors help the robot to perceive its environment. The detection range is between 5 and 2 cm. Sensor data are real values between 0.0 (nothing in front) and 1.0 (obstacle nearby), each data is coded on ten bits. All the measurements depend largely on various factors like the distance from the obstacle, the color, the reflectance, the vertical position, etc. The computational power of the robot is equivalent to a M68030. Energy autonomy is thirty minutes.

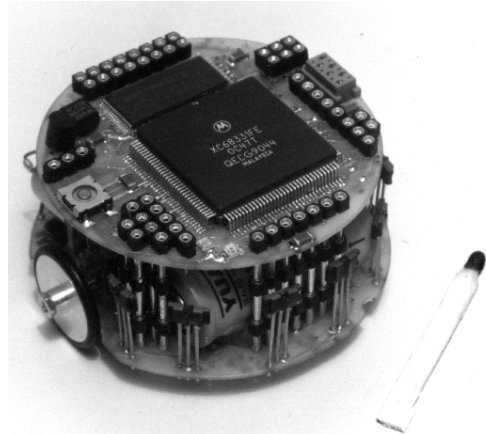


Fig. 1 The robot Khepera

3. The reinforcement function

The task Khepera must perform is to move forward when possible and avoid obstacles. Two behaviors are involved. One, with a higher priority, moves forward and a second proposes avoidance actions. The first behavior is so simple, i. e. move forward when nothing is detected by the sensors, that it is of no interest here. On the contrary, the second involves knowing how much to turn and in which direction so as to avoid the obstacles. The environment is an arena of A4 size. Obstacles are film cans or whatever is suitable (lipstick, eraser, etc.). Obstacles are put in the arena at the beginning of the experiment or during the learning phase.

The robot receives the following reinforcement signals for avoiding:

- +1 if it is avoiding, or
- 1 if a collision occurs, or
- 0 otherwise.

The robot is avoiding when the present sum of sensor values is smaller than the last one, the difference been superior to 0.06. A collision occurs when the sum of the six front sensor values is superior to 2.90 or the sum of the two back sensor values is superior to 1.95. Threshold values like (0.06, 2.90, 1.95) have been determined after extensive experiments.

4. Experiments with a self-organizing map implementation

A self-organizing map (SOM) [4] is used to store the Q-values. The learning phase associates to each neuron of the map a situation-action pair plus its Q-value. The number of neurons equals the number of stored associations. After some experiments, a size of sixteen neurons is selected. There are 176 connections (11 x 16).

In all experiments, Khepera uses a random number generator to control the randomness of action selection. The randomness decreases proportionally to the inverse of the number of iterations. It is a crude strategy for active exploration, but sufficient for our experiments.

Results

The usual way of presenting the performance is to plot the cumulative reward over time. It is the number of correct actions performed from iteration 1 to iteration t divided by the number of iterations (t) [5]. An action is considered correct if it does not generate a collision. Using only the reinforcement function given above, Fig. 2 displays three different trials: after 500 learning iterations the behavior is correct.

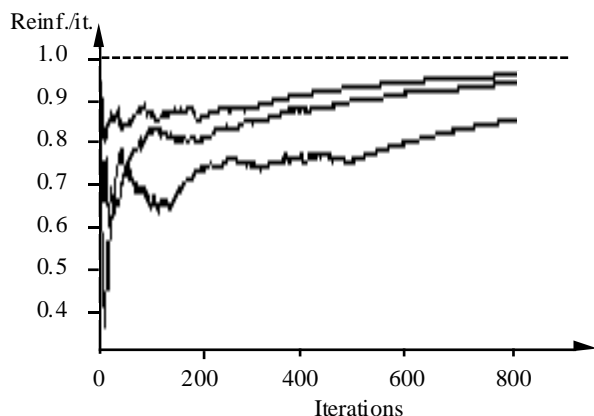


Fig. 2 Three self-organizing map learning curves. The first 500 iterations correspond to the learning phase.

The topology preserving mapping property of the self-organizing map allow to verify that the behavior is correct. During the learning, the Q-values become positive for all the neurons: the actions undertaken in the encountered situations are correct. Fig. 3 displays the mean

of the Q-values of all the neurons during the learning phase: the global performance of the robot is increasing.

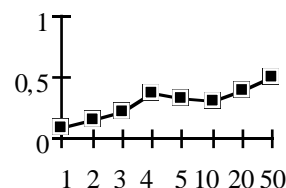


Fig. 3 Mean Q-values of all neurons during the learning phase. The global performance is increasing: the resulting behavior is correct. The number of iterations has to be multiplied by 10.

5. Diversity of the obtained behaviors

However, despite positive Q-values for all actions, the obtained behavior does not always express the expected behavior. For example, the learned behaviors show a large distribution of covered distances. Fig. 4 displays four curves corresponding to four different experiments. Behaviors may display a predilection for forward moving, or for backward moving, or for small movements, or for change in their policies during the experiment.

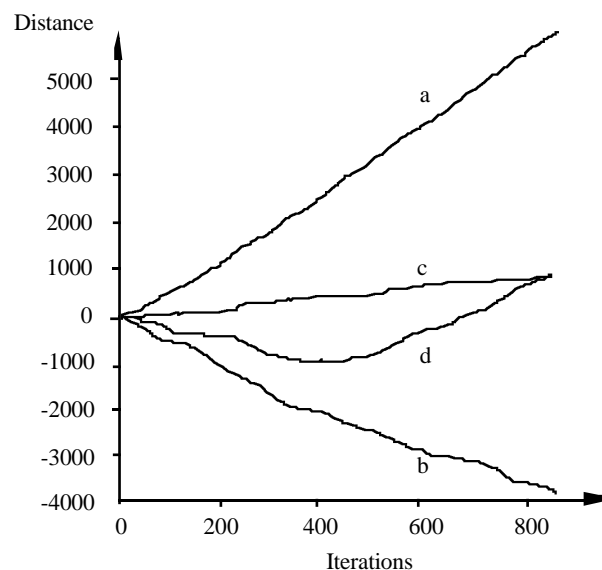


Fig. 4 Distances covered by Khepera during four different experiments of learning an obstacle avoidance behavior. Behavior (a) displays a predilection for forward moving, (b)

prefers backward moving, (c) prefers small forward movements, (d) changes its policy at the end of the learning phase (500 iterations).

There was also another possible behavior, but it was so annoying that we have prefer to forbidden it mechanically. It is moving forward to an obstacle, stop before the sum of the sensor values is superior to the threshold and then moving backward, and doing it all again and again. This sequence of actions maximizes rewards, but it is too far from an obstacle avoidance behavior.

It is quite impossible to modify the reinforcement function so to take into account things like: not doing an action and then its opposite, or not doing too many small movements, or not doing only backwards avoidance. This comes from the fact that there is no concept of state in the SOM implementation of Q-learning. The only way to deal with time delay is due to the Q-learning [6] updating equation (1), but this equation solves only part of the temporal credit assignment problem. Its efficiency is limited on a large sequences of actions.

$$Q(x, a)_{\text{new}} = Q(x, a)_{\text{old}} + \beta(r + \gamma \cdot \text{Max}_{a'} Q(x', a') - Q(x, a)_{\text{old}}) \quad (1)$$

where a is an action, x is the input state, x' is the state after executing a in x , r is the immediate reward, β and γ equals 0.5 and 0.9 respectively.

6. Forbidden sequences of actions

We propose to improve the learning with the use of a set of forbidden sequences of actions. We indicate for the three mentioned problems the corresponding sequences of actions to forbid.

1/ Moving back and forth -> Alternated sequences of actions having the same absolute values.

2/ Small movements -> long sequences of actions having small absolute values.

3/ Backward avoidance -> Long sequences of actions having negative values.

All these sequences modify the exploration function. The effect is to suppress the eligibility of actions in a given situation (and a given historical context). On the SOM implementation, the second closest neuron is selected instead of the first.

7. Experiments

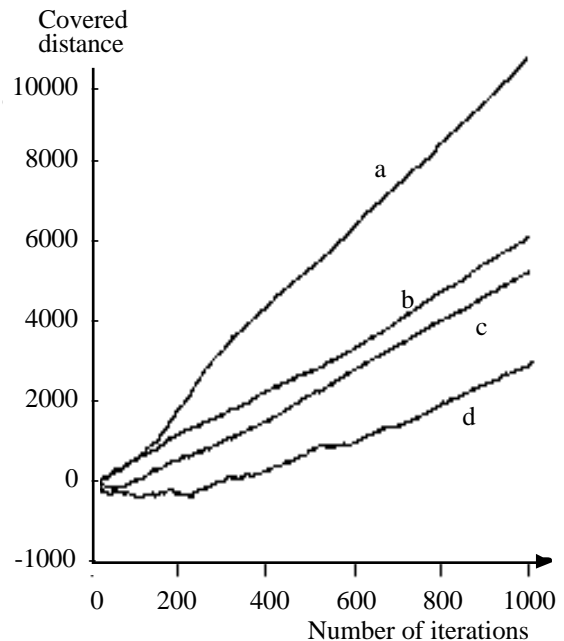


Fig. 5 displays the covered distances of four experiment (a, b, c, d). They are different, depending on the initial random conditions.

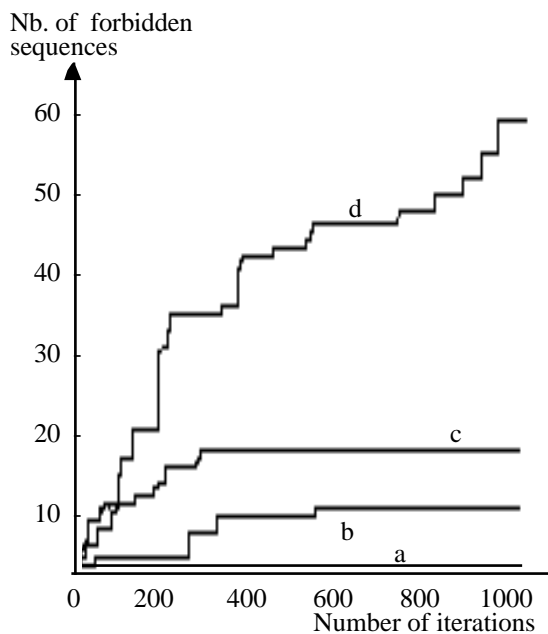


Fig. 6 shows the number of forbidden sequences used per experiment. Not all experiment need a large implication of this module, but it can be necessary, i. e. experiment (d).

We ran several experiments, which point out that only forward behaviors were learned (Fig. 5). The number of the forbidden sequences used per experiment is reported on Fig. 6. It shows that not all experiment need a large implication of this module. To a first look, it seems that the involvement of the forbidden sequence module is proportional to the inverse of the covered distance. The truth is different. The learned behavior starts from initial random conditions. The less the behavior favorites long covered distances, the more the involvement of the forbidden sequence module. This is particularly visible in experiment (d).

8. Conclusion

In this paper, we have proposed an improvement of reinforcement learning through the use of forbidden sequences of actions. A given reinforcement function can generate a large set of behaviors, all with good performances. But, usually only a small subset corresponds to the expected behavior. It may

be difficult to tune the reinforcement function, so we have add an external module containing generic forbidden sequences of actions. It is a way for the combinational implementation of Q-learning to deal with time delay. Experiments with the real miniature robot Khepera in a task of learning an obstacle avoidance confirm the interest of this approach.

Our experiments involve no speed-up techniques other than generalization of the self-organizing map. It would be interesting to test the impact of forbidden sequences on other methods like experiment replay, action models, teaching [7], Dyna model [8], generalization with Hamming distance or clustering [9], etc.

9. Acknowledgments

We thank all the K-Team members (LAMI-EPFL, Switzerland) for their interest in this research and the use of one of the first Khepera robots.

10. References

- [1] C. Touzet, "Neural Implementations of Immediate Reinforcement Learning for an Obstacle Avoidance Behavior," submitted to IEEE-SMC, special edition on Learning Approaches to Autonomous Robots Control, M. Dorigo guest editor.
- [2] D. Ackley and M. Littman, "Interactions Between Learning and Evolution," *Artificial Life II*, SFI Studies in the Sciences of Complexity, vol. X, C. G. Langton & Co. Eds, Addison-Wesley, pp. 487-509, 1991.
- [3] F. Mondada, E. Franzi and P. Ienne, "Mobile robot miniaturisation: A tool for investigation in control algorithms," *Third International Symposium on Experimental Robotics*, Kyoto, Japan, October 1993.
- [4] T. Kohonen, *Self-Organisation and Associative Memory*, Springer-Verlag, Vol. 8, Berlin, 1984.

[5] M. Dorigo and M. Colombetti, "Training Agents to Perform Sequential Behavior," *Adaptive Behavior*, MIT Press, 2 (3), pp. 247-276, 1994.

[6] C. J.C.H. Watkins, "Learning from Delayed Rewards," PhD thesis, King's College, Cambridge, 1989.

[7] L-J. Lin, "Reinforcement Learning for Robots Using Neural Networks," PhD thesis, Carnegie Mellon University, Pittsburgh, CMU-CS-93-103, January 1993.

[8] R.S. Sutton, "Reinforcement Learning Architectures for Animats," Proceedings of the First International Conference on Simulation of Adaptive Behavior, *From Animals to Animats*, Edited by J-A Meyer and S.W. Wilson, MIT Press, pp. 288-296, 1991.

[9] S. Mahadevan and J. Connell, "Automatic Programming of Behavior-based Robots using Reinforcement Learning," *Artificial Intelligence*, 55, 2, pp. 311-365, July 1991.