BEHAVIOR EVALUATION AND LEARNING FROM AN INTERNAL POINT OF VIEW

François Michaud and Maja J. Matarić
Interaction Lab, Volen Center for Complex Systems
Computer Science Department
Brandeis University
Waltham, MA 02254-9110
tel.: (617) 736-2738 (617) 736-2708 fax: (617) 736-2741
{michaudf,maja}@cs.brandeis.edu, http://www.cs.brandeis.edu/~agents

Abstract

This paper describes a new approach that makes a robot learn by evaluating its own performance based on the use of its resources. For a behavior-based robot, this means that learning is accomplished from the observation of behavior use over time. The acquired knowledge can then be exploited for future selection of behaviors. When applied to the multi-robot domain, this approach will make the robot find regularities in its interactions with its environment and exploit them efficiently, eventually resulting in specialization within the group.

1 INTRODUCTION

Learning can be used in many ways to improve the ability of a robot to behave in its environment, such as by acquiring knowledge about the world (e.g., using a topological representation) [6, 9], by acquiring special skills (behaviors or control policies) [4, 10] or by learning when to use such skills according to sensed conditions [3, 7].

Learning in the mobile robot domain is challenging. First, there are difficulties intrinsic to the robot, caused by incomplete and imprecise perception of the environment and by limited processing and memory capabilities. These difficulties limit the robot's ability to represent the world and to evaluate the events that take place in the environment. Second, the learning process itself brings other types of problems. For example, reinforcement learning typically requires high numbers of training trials along with stationary conditions in the environment. These difficulties are especially present in the multi-robot domain where the dynamics of the environment are particularly important, making the learning task even harder.

Considering these facts, one approach to more effective learning could be to make the robot evaluate performance according to its own perspective, i.e., based on the use of its resources. Learning the optimal solution would then be defined according to the internal point of view of the robot, instead of being evaluated from an external observer. For a behavior-based system [1], resources are the behaviors available for controlling the actions of the robot. Assuming that behaviors are programmed with an initial policy, the objective is to try to find some regularities in the interactions between the robot and the world reflected in the robot's use of behaviors over time. This knowledge can then be exploited for improving the way the robot behaves by affecting the future selection of its behaviors.

The purpose of this paper is to present such an approach for mobile robot learning in a multi-robot domain. The paper is organized as follows. Section 2 justifies the need to evaluate performance from the point of view of the robot. Section 3 describes the learning algorithm for behavior-based system that we are developing based on these ideas. Section 4 overviews the related work and Section 5 concludes the paper.

2 SELF-EVALUATION FOR LEARNING

The performance of a robot situated in dynamic, noisy environments is difficult to evaluate using traditional, externally-measured criteria. If the behavior of such a robot changes over time due to learning, the problem of external evaluation is even more complex. This is the case at least in part because typical evaluation criteria were developed for and adapted from very different, largely more static and predictable domains. In contrast, the performance of a learning

0

robot results from the interaction dynamics at a variety of levels. This issue is particularly acute in the multi-robot environment we are studying, where interactions occur between behaviors within each of the robots, between a robot and the objects in the world (blocks or obstacles), and between two or more robots. Since all of these interactions have an impact on the overall "behavior" and "performance" of the system, it is difficult to combine them into an external numerical evaluation function.

We propose a method that allows the system to monitor its own performance and employ an internal estimate to alter its behavior selection. Consequently, the system is able to adapt to the changing dynamics without the user having to predict those dynamics *a priori*. For example, in the foraging case, the user cannot easily predict at what point in time and space interference patterns will significantly peak and alter. These dynamics, however, can be detected by the robots themselves, and used to adapt and improve performance over time. Thus perhaps the most workable solution from both an engineering and evaluation perspective is to initially endow the system with some knowledge, i.e., some initial policy, and then allow the system to further refine and alter that policy over time, as the dynamics change. It is important to enable the system to adapt in a timely fashion; the idea we are proposing is focused on continuous life-time adaptation rather than on learning a one-time, static policy.

3 LEARNING FROM HISTORY OF BEHAVIOR USE

First, we assume that a robot is given a repertoire of behaviors by the designer. These behaviors define the robot's skills for handling the situations encountered in its environment. In a behavior-based system, behaviors from the repertoire all run in parallel and are usually prioritized using an arbitration mechanism like subsumption [1]. However, allowing all behaviors to concurrently take part in the control of the robot might be impossible when the robot has multiple tasks to accomplish and needs to behave differently depending on its changing goals. A possible solution is to dynamically activate the appropriate behaviors according to the robot's goal [7, 9]. In that regard, knowing when to use a behavior is as important[1] as the behavior itself. We assume than conditions to activate behaviors can also be pre-programmed by the designer according to what can be anticipated about the tasks the robot needs to accomplish.

[1] and maybe more

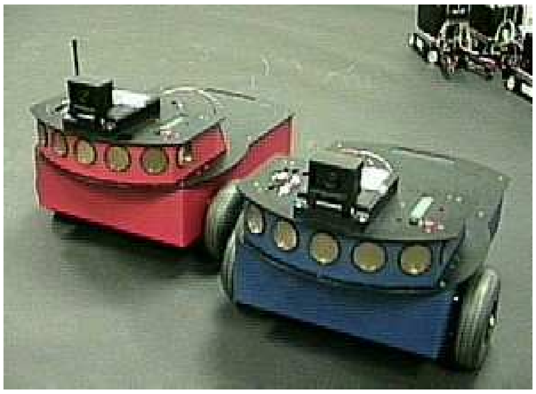

Figure 1: Pioneer robots used in the foraging experiment, Ben and Mae.

Behaviors can be activated according to sensed conditions from the world, and it has been shown that these can be learned through reinforcement learning [3, 7]. But observing behavior use over time can also be an interesting source of information for behavior activation [9], especially in the multi-robot domain where the dynamics with the environment are very important. Using this information, we want the robot to learn to activate behaviors based on the history of their use. By evaluating behavior sequences, the robot can draw on past experiences to anticipate what can happen in the future and try to improve the overall performance by reproducing conditions that were, in its view, more efficient. The next subsections describe an algorithm that we propose based on this idea, explained in the context of a foraging experiment in a multi-robot domain.

3.1 Learning when to activate alternative-behaviors

For implementing the learning algorithm, we are using Pioneer 1 mobile robots (Figure 1). These robots are equipped with seven sonars and a Fast Track Vision System. They are programmed using MARS (Multiple Agency Reactivity System) [2], a language for programming multiple concurrent processes and behaviors. In the experiments, one Pioneer is placed in a "corral" with another Pioneer and other robots, and its goal is to find blocks and to bring them home. In doing so, the Pioneer may experience interference from the presence of the other robots in the environment. Blocks and home are of distinct colors and are detected by the robot's vision system.

As indicated earlier, each Pioneer is given a set of behaviors and the activation conditions of some of them in regard to its goal. These behaviors can be classified according to their respective purposes, as follows. First, the Pioneer has behaviors for accomplishing the task of finding and grasping blocks, such as *wandering* and *grasp-block*, and for the task of bringing them home, such as *homing* and *drop-block*. These behaviors are referred to as task-behaviors (tb). Conditions for activating task-behaviors are given according to the presence or absence of a block in front of the robot and to the location of the robot relative to home. Second, behaviors such as *avoidance* and *backup* are given to the Pioneer for handling harmful situations and interference experienced while accomplishing its tasks. These behaviors are referred to as maintenance-behaviors (mb). Using this preprogrammed knowledge, the Pioneer is able to bring the blocks home with varying efficiency, which is characterized by the amount of time required to accomplish the task and the interference experienced during that process. This second factor is reflected in the use of the maintenance-behaviors while accomplishing the task, indicating that events unrelated to the goal must be handled based on priority. For example, if the Pioneer has to go through a crowded area, then its time will be mostly spent avoiding others instead of going toward home, causing it to be less efficient.

Third, for improving the robot's performance, the designer provides the Pioneer with a set of alternative-behaviors (ab) that may be useful in changing the way a task is achieved. No *a priori* conditions are given for their activation: the objective is to let the robot learn when to activate these alternative-behaviors in accordance to its own evaluation of efficiency and its own experiences in the world. For example, the Pioneer may decide that it is better to do *wall-following* or *rest* for a little while when experiencing some difficulties in accomplishing its task. This might lead to interesting group behavior learned from the point of view of the robot, whose exact conditions need not be specified or even anticipated by the designer.

3.2 Learning algorithm based on behavior use

The learning algorithm proposed in this paper is composed of two parts: 1) a metric to evaluate performance from the history of behavior use, and 2) a mechanism for representing the sequence of their use. The metric that we propose evaluates the ratio of the time spent not using maintenance-behaviors compared to the total amount of time spent accomplishing the task. This reflects the amount of control the robot has with regard to the task. Relation 1 expresses that measure of efficiency, where a time unit corresponds to one cycle of evaluation of all currently active behaviors.

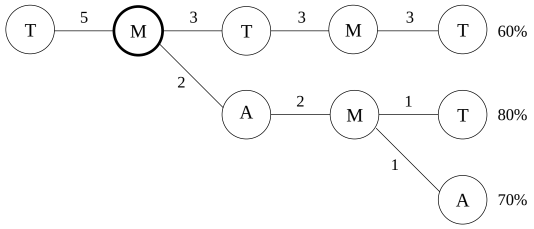

Figure 2: Graph representation of the sequences of behaviors use (T is for task-behaviors, M is for maintenance-behaviors, and A is for alternative-behaviors)

$$eff(t) = \frac{(t_{tb} - t_{mb})}{t_{tb}} \tag{1}$$

We use a graph as the representation mechanism to encode the history of behavior use for a particular task (such as searching for a block or bringing a block home). Figure 2 shows a typical graph in which nodes represent the type of behaviors used for controlling the robot while accomplishing its task, and links store the number of times a transition between two types of behaviors was observed. The sequence of nodes of a path characterizes the interactions experienced by the robot in the world, and situate the robot in reference to these interactions. Initially, the graph for a particular task is empty and is incrementally constructed as the robot goes about its task. When an alternative-behavior is applied, its name becomes the node type to characterize the choice made by the robot at that point in the graph. Similar paths can be followed according to the sequence of behavior use and the nodes stored in the graph. If a complete path is reproduced by the robot, the average of the stored and current efficiencies is computed and the stored value is updated.

While the robot is accomplishing a task, the graph representation is also used to compare the current efficiency with the efficiencies experienced in the past to anticipate the future outcomes of its choices. To do this, the robot evaluates the expected efficiency from the current position in the graph. This evaluation occurs only at a maintenance-behavior node because that is when *eff(t)* decreases. The expected efficiency from any point in the graph is the sum of the memorized efficiencies in connected paths multiplied by their probability of occurrence. These probabilities are derived from the frequency of reproduction of the paths relative to the current position in the graph. For example, if the current position in Figure 2 is the bold node, then the expected efficiency for the subpath starting with the task-behavior is $(60 * 3/5)\%$ while the expected efficiency for the subpath using an alternative-behavior is $((80 + 70) * 1/5)\%$. The total expected efficiency for this node is the sum of the expected efficiencies or these subpaths, i.e., 66%.

From these measures, the robot has to decide whether to continue without using an alternative-behavior, to use an untried alternative-behavior, or to use an alternative-behavior that was previously tried. The selection criteria are based on the current efficiency and the options already tried at the current position in the graph: if the current efficiency is lower than the expected efficiency, then the robot decides to try an alternative that was not tried before at this point in the graph. Otherwise, the robot favors the option with the highest efficiency. Note that when a new path is constructed, its expected efficiency is 0 and the favored option is to behave without using any alternative-behavior.

At the present time, this learning algorithm is implemented on the Pioneer and we are in the process of testing and gathering data. Because the evaluation of the performance is done from the point of view of the robot, evaluating the learning algorithm is done according to the graph stability and adaptability in the situations occurring in the environment. The actual utility of the graph will be analyzed by evaluating its effect on the robot's behavior selection. Finally, we plan to evaluate the effects on the graph (depth, breadth) according to different efficiency measures and selection criteria.

4 RELATED WORK

To the best of our knowledge, there has not been much work done in the context of associating a sequence of events, reflected in behavior use, to influence future activation of behaviors. Related work includes Ram and Santamaria [11] who consider history from inputs and from behavior activation for setting future behavior activation. A case-based algorithm is used to retrieve similar cases, adapt the activation of behaviors, learn new associations or adapt existing cases. A reward signal is used to try to maximize consistency in "useful" behaviors. Problems related to this approach are its large processing time, the difficulty to interpret the regularities captured in the system's cases, the necessity of setting a lot of parameters, and the fact that this action model was not used either during or subsequent to its acquisition.

The work of McCallum [8] is most related to ours in the notion of learning from history. McCallum uses a tree representation in a reinforcement framework to learn the appropriate actions to take according to the relevant features and the memory of those features from previous states. This mechanism is successfully applied to learning to drive in a virtual highway driving environment, but has not been tested in the robot domain.

5 CONCLUSION AND CONTINUING WORK

This paper presents the idea of making the robot learn from self-evaluation of its own resource use. A learning algorithm for behavior-based systems [1] is proposed based on this idea. In this learning algorithm, the use of behaviors over time is represented in a graph, and performance is evaluated according to the purpose of the behaviors used. This graph is continuously updated and used to anticipate the expected efficiency of the robot in accomplishing its task, and for selecting different behaviors when striving for performance improvement. Using this simple algorithm, we hope that the robot will be able, from its own perspective, to find regularities in the interactions with its environment and to learn from them.

In addition to introducing new insights from the point of view of an individual and its interactions in a multi-robot domain, this mechanism can provide a predictive capability that is lacking in behavior-based system, thus making a move toward increasingly cognitive means for generating complex behavior. This project also tackles important problems of incomplete and imperfect perception, learning using prior knowledge and multi-agent learning [5]. We are currently in the process of testing this learning algorithm in a multi-robot foraging task, and we hope to get a better understanding of its effects on group dynamics.

Acknowledgments

The authors acknowledge the financial support of the NSERC (Natural Sciences and Engineering Research Council of Canada), the Office of Naval Research Grant N00014-95-1-0759 and the National Science Foundation Infrastructure Grant CDA-9512448 for this research. Many thanks to our robot photographer, Barry Werger.

References

[1] Rodney A. Brooks. A robust layered control system for a mobile robot. *IEEE Jour-*

nal of Robotics and Automation, RA-2(1):14–23, March 1986.

[2] Rodney A. Brooks. Mars: Multiple agency reactivity system. Technical report, IS Robotics, 1996.

[3] Pattie Maes and Rodney A. Brooks. Learning to coordinate behaviors. In *Proc. Heigth National Conf. on Artificial Intelligence AAAI*, volume 2, pages 796–802, 1990.

[4] Sridhar Mahadevan and Jonathan Connell. Automatic programming of behavior-based robots using reinforcement learning. *Artificial Intelligence*, 55:311–365, 1992.

[5] Sridhar Mahadevan and Leslie Pack Kaelbling. The NSF workshop on reinforcement learning: summary and observations. *AI Magazine*, December 1996.

[6] Maja J. Matarić. Integration of representation into goal-driven behavior-based robots. *IEEE Trans. on Robotics and Automation*, 8(3):304–312, 1992.

[7] Maja J. Matarić. Reinforcement learning in the multi-robot domain. *Autonomous Robots*, 4(1), January 1997.

[8] Andrew Kachites McCallum. Learning to use selective attention and short-term memory in sequential tasks. In Pattie Maes, Maja J. Matarić, Jean-Arcady Meyer, Jordan Pollack, and Stewart W. Wilson, editors, *From Animals to Animats 4: International Conference on Simulation of Adaptive Behavior*, pages 315–324. The MIT Press, 1996.

[9] François Michaud, Gérard Lachiver, and Chon Tam Le Dinh. A new control architecture combining reactivity, deliberation and motivation for situated autonomous agent. In Pattie Maes, Maja J. Matarić, Jean-Arcady Meyer, Jordan Pollack, and Stewart W. Wilson, editors, *From Animals to Animat 4: International Conference on Simulation of Adaptive Behavior*, pages 245–254. The MIT Press, 1996.

[10] Ulrich Nehmzow, Tim Smithers, and Brendan McGonigle. Increasing behavioural repertoire in a mobile robot. In Jean-Arcady Meyer, Herbert L. Roitblat, and Stewart W. Wilson, editors, *From Animals to Animats 2: International Conference on Simulation of Adaptive Behaviors*, pages 291–297. The MIT Press, 1992.

[11] Ashwin Ram and Juan Carlos Santamaria. Multistrategy learning in reactive control systems for autonomous robotic navigation. *Informatica*, 17(4):347–369, 1993.