Combining Attention and Value Maps

Stathis Kasderidis[1§], John G. Taylor[2]

[1]Foundation for Research and Technology – Hellas
Institute of Computer Science
Vassilika Vouton, P.O. Box 1385, 711 10 Heraklion, Greece
stathis@ics.forth.gr
[2]King's College, Dept. of Mathematics,
Strand, London WC2R 2LS, UK
john.g.taylor@kcl.ac.uk

Abstract. We present an approach where we combine attention with value maps for the purpose of acquiring a decision-making policy for multiple concurrent goals. The former component is essential for dealing with an uncertain and open environment while the latter offers a general model for building decision-making systems based on reward information. We discuss the multiple goals policy acquisition problem and justify our approach. We provide simulation results that support our solution.

1 Introduction

There is currently much research interest in developing autonomous agents. One of the primary problems in the field is that of multiple goal satisfaction. Approaches such as reinforcement learning have provided a general method for modelling the goal satisfaction problem for the case of a single goal at a given time [1]. Finding a suitable policy for multiple concurrent goals is a generalisation of the previous problem. Again reinforcement learning methods can be applied directly. However, the approach lacks the ability to deal with an ever-changing environment in an immediate way. This can be handled by attention. There is much work in recent years that has been devoted in understanding attention [2]. It has been modelled in a control theory framework by the second author [3]. Inspired by the above developments we have developed recently the Attentional Agent architecture [4] which combines a goals-based computational model with an attention mechanism for selecting priority of goals dynamically in run time, for handling novelty and unexpected situations as well as learning of forward models based on the level of attention [5]. We now extend this model further to allow the attention mechanism to act as an alarm system when we approach limiting conditions. This model can be combined with a reinforcement learning approach for single goals to provide the solution for multi-goal policy acquisition. The structure of the paper is as follows: In section 2 we present a concrete problem statement and we describe the environment used for testing a robotic agent. In section 3 we review the Attentional Agent Architecture and its extension to multi-goal policy acquisition. In section 4 we present supporting simulations.

§ Corresponding author.

2 Problem Specification

To test the proposed architecture we select a robot navigation task. The concrete setting is as follows: We assume the existence of a suitable space where the robotic agent moves from a point A to a point B transporting some item of interest. The high-level decomposition of the task is shown in Fig. 1. Transport is the primary agent goal. At the same time there is the goal of maintaining itself in a working condition, which appears in Fig. 1, as the Power Monitor goal. Inside the space there are a number of stationary and moving objects. The overall task is to provide the Transport service of moving items from point A to B while avoiding collisions with other moving and stationary objects and also by making sure that the robot always has enough power. If the power level drops low the agent should recharge itself and return to the previous task. Recharging can take place in the recharge station (Terminal C) or by collecting small charges from objects of class Obj+ (by moving to the same grid cell). Correspondingly, objects of class Obj- should be avoided as they reduce the power level if touched. We assume that objects Obj+ carry a reward of +1, while objects of Obj- carry a reward of -1. Moving to goal position B achieves a reward signal of +20. All other states are assumed as having zero reward initially. A possible configuration of the space is shown in Fig. 2.

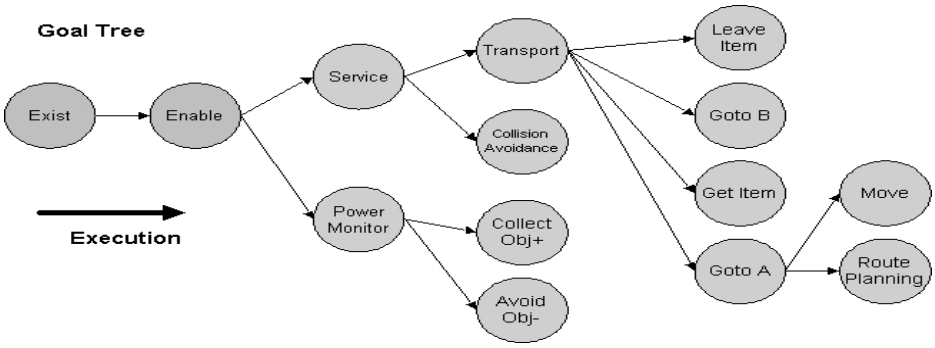

Fig. 1. Goals Tree of agent.

In Fig. 1 there are three high-level goals: Transport, Collision-Avoidance and Power Monitor. The Transport goal executes a sequential program of four sub-goals {Goto A, Get Item, Goto B, Leave Item}. The Goto A goal is further decomposed to Route-Planning and Move goals. All goals are executed ultimately by calling primitives which are not shown in Fig. 1. The Route-Planning goal is responsible for collecting the current sensory state and for calculating a new position for moving the agent closer to the goal position, based on the value map of the corresponding goal. Then it passes this new location to the Move goal to execute it. Internally it uses predictive models (Forward Models) for forecasting the possible position of the other moving agents.

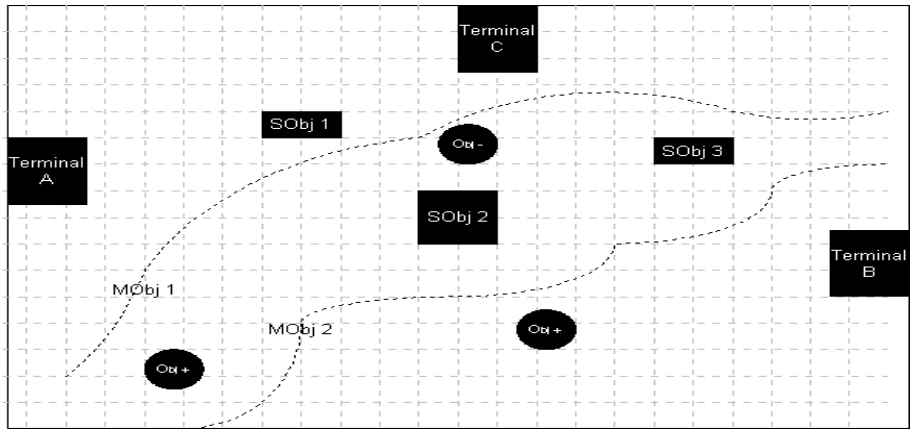

Fig. 2. A possible configuration of GridWorld. MObj x represent a moving object x. SObj x represent static objects. Obj+/- represent objects that are assumed static but having a positive/negative influence on the power level of the agent if touched. Terminal C is the recharging station. Terminals A and B is the start and finish position of the Transport goal.

Given that a location is "closer" to the target position and it will not be occupied by other agents in the next time instance, the location is selected. During the execution of the actual move, it might occur that some other agent moved to the calculated location, because our movement prediction was wrong. We guard against such a case using the Collision-Avoidance goal, which implements a motor attention scheme. This goal is normally suppressed by the Transport goal; if however a collision is imminent, an attention event is created, which in turn raises the overall Action-Index of the goal. The final result is that the Transport goal is suppressed due to losing the global attentional competition against the Collision-Avoidance goal. The Power Monitor goal monitors the current power level and if low it will re-direct the robot to the recharging terminal or to a nearby Obj+ object. The policy is not hard-wired but learnt as described in section 3.2. Care is given to avoid objects Obj-. Collision with a static or moving object corresponds to a reward of -5 and -10 respectively. We also assume that the maximum power level is 1000 power units, motion expends 1 power unit per cell, and touching Obj+ and Obj- increases / reduces the power level by +10/-10 units with a reward of +1/-1 respectively.

3 Multiple Goal Policy Acquisition for Attentional Agents

3.1 Attentional Agent architecture

The Attentional Agent architecture was thoroughly discussed in [4]. An Attentional Agent is a system which has the following major components: 1. A goal set, organised in a tree (GoalsTree), see Fig 1; 2. A complex execution unit, called a goal, with an internal structure; 3. A global attention-based competition mechanism, which in-

fluences the priority of the goals; 4. A local attention-based mechanism, in the scope of a goal, which detects novel states and initiates learning of new models or adaptation of existing ones. The local attention process works in dual modes: sensory and motor ones; 5. Each goal contains the following major modules: *State Evaluation*, *Rules*, *Action Generation*, *Forward Models*, *Observer*, *Attention* (local) *Controller*, *Monitor*, *Goals.* The first five modules are implemented by models which can be adapted if erroneous performance is realised. We consider here that the Rules module is a Value Map that is created through reinforcement learning; 6. Partitioning the input and output spaces into suitable sub-sets. The input space is the sensory space, while the output space is the action space. We extend this model to include in the local scope an additional attention process that of the *Boundary Attention*. This process is responsible for raising an attention event when we approach a limiting condition in the scope of a goal. For example when the power level is low this can be considered as a boundary condition that must capture attention and thus increase the priority of the goal. This relates to general *homeostasis* mechanisms of biological agents. When a homeostatic variable moves near (or out of) the boundary of its preferred range then attention is raised so that appropriate corrective action will be taken.

3.2 Policy acquisition for multiple goals

Our proposed solution for multiple goals policy acquisition is based on the following ideas: i. Use of attention allows the agent to have fast reaction and deal with novel and unexpected situations that develop in a time scale faster than that used for single-goal policy learning; ii. Instead of learning an overall (joint) policy for the current set of goals directly it is simpler to combine individual goal policies to an overall strategy. Learning an overall policy seems unlikely in biological agents as one has to store a value map (of the policy) for each combination of goals ever encountered; iii. We effectively acquire an overall policy by selecting at each time instance only one active goal and the action output of the agent is the action selected in the scope of the active goal. The learning of the goal's policy takes place in an individual basis using a standard RL method; iv. The scheme for selecting priorities for a set of competing goals is based on the following formula:

$$\text{ActInd}=(W+\text{S-AI}+\text{M-AI}+\text{B-AI}+\sum\text{ActInd}) / (4 + \#\text{ Contributing Children}) \quad \textbf{(1)}$$

Formula (1) is an extension of the corresponding formula in [4]. ActInd is the action index of goal, which effectively controls the priority of the goal in global competition. W is the "intrinsic" weight. S-AI is the sensory attention index (to capture novel and unexpected situations), M-AI is the motor attention index (to capture impending dangers), B-AI is the boundary attention index discussed in 3.1. All attention indices and the intrinsic weight are bounded in [0,1]. The sum of Action Indices in the numerator is over all contributing children in the sub-tree of a goal. A child is contributing if any of its corresponding attention indices is non-zero. This allows for the propagation of attention events in the goal hierarchy; v. Non-competing goals (due to referring to a different sub-region of the action space) are processed in parallel.

4 Simulation Results

We use the setting described in section 2, the set of goals in Fig. 1, formula (1) for selecting goal priorities and we define the sensory and motor attention indices as in [4,5]. The boundary attention index is defined as a sigmoid function over the value map of the energy states of the agent, and it is given by (2):

$$\text{B-AI}=1/(1+\exp[\ \text{Val}(E(t)-E_{max}/2)\ ]\ ) \quad \textbf{(2)}$$

where Val(·) is the value for the corresponding energy state (as the sum of all future discounted rewards) – it takes negative values for negative energies -, E_{max} is the maximum energy level (1000 units) and the energy value map has been acquired as all other value maps using the Q-learning algorithm [6]. The general methodology for training was as follows: We first trained the agent in each sub-problem separately using the Q-learning method with parameters of a=0.1 and γ=0.9 for learning and discount rate respectively. Then we allowed the existence of multiple goals concurrently. The selected action at each time step was determined by the currently active goal by using its own value map acquired during the separate training. The active goal is the goal which wins the attentional competition. The maximum number of training sessions for learning a value map was a million iterations. The intrinsic weights in (1), which code the relative importance of goals are given as relative ratios: $|Val_G|/\sum |Val_G|$ of the values for each goal, in our case: +20 for Transport (point B), +10 for Power Monitor (point C), -10 for MObj collision, -5 for SObj collision, +1 for Obj+ collision, -1 for Obj- collision and 5 for Collision Avoidance respectively. When approaching one of SObj, MObj, Obj+, Obj- closer than a threshold range R=3 cells, we calculate the S-AI and M-AI indices as described in [4,5]. Adding their contribution to (1) allows the alternation of priorities of goal execution and thus the determination of the currently active goal. The size of the GridWorld was 50x30 cells. We assume that when the agent reaches point B it then returns to A for starting another Transport action. It continually expends energy. We run 50 simulations to check the probability of collisions and switching from the Transport goal to the Power Monitor goal. Each simulation session included the execution of 10 Transport commands (so as to deplete the energy and force a recharge). The results show that the agent successfully switched to the appropriate goal's value map for action selection in all cases. In some cases collisions with moving objects took place due to false predictions regarding the future position of the objects. With further training of the predictive models for the motion of moving objects, the collisions are decreased, as it was also described in [5]. We used from 5-15 moving objects having different paths per simulation as in [5]. The overall performance for the agent is shown in Fig. 3. We show a curve that depicts the ratio of lengths of path A-B-C-A for the best/current policy against the number of training iterations for acquiring the policy (250K, 500K, 750K and 1M). The "best" policy was determined by us empirically using the optimal action at each time step. The curve is drawn by using the total length of the overall curve from point A to B and back. We used only the curves during which we had a recharge event and we averaged the lengths over the 50 simulation sessions. It is clear that as the learnt individual goal policies approach their

optimal targets the actual length comes closer to the shortest length. In Fig. 3 all policies corresponding to goals were trained to the same level.

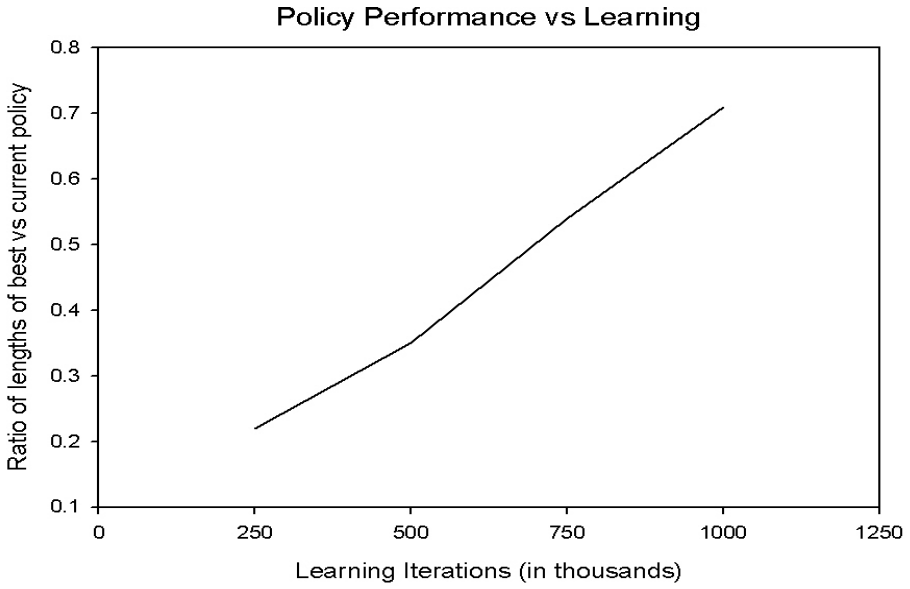

Fig. 3. Ratio of path lengths (A-B-C-A) of "best" vs current overall policy against the number of iterations for learning a policy. The path includes a *recharge* event and the current policy lengths are averaged over 50 simulations.

Acknowledgments

We would like to acknowledge the support of the European Union through the FP6 IST GNOSYS project (FP6-003835) of the Cognitive Systems Initiative to our work.

References

1. Sutton R. & Barto, A. (2002). Reinforcement Learning: An Introduction, MIT Press 2002, 4th Ed.
2. Rushworth M. F. S. et al (1997). The left parietal cortex and motor attention. Neuropsychologia 33, 1261-1273.
3. Taylor J. G. (2000). Attentional movement: the control basis for Consciousness. Soc. Neuroscience Abstracts 26, 2231, #893.3
4. Kasderidis, S., & Taylor, J.G. (2004). Attentional Agents and Robot Control, International Journal of Knowledge-based and Intelligent Systems 8, 69-89.
5. Kasderidis, S. & Taylor J.G. (2004). Attention-based Learning, International Joint Conference on Neural Networks (IJCNN 2004), Budapest 25-29 July 2004, p.525-531.
6. Mitchell T. (1997). Machine Learning, McGraw Hill.