A Self-Made Agent Based on Action-Selection

Olivier Buffet BUFFET@LORIA.FR
Alain Dutech DUTECH@LORIA.FR
LORIA / INRIA-Lorraine, BP 239 - 54506 Vandœuvre-lès-Nancy - France

Abstract

Some agents have to face multiple objectives simultaneously. In such cases, and considering partially observable environments, classical Reinforcement Learning (RL) is prone to fall in pretty low local optima, only learning straightforward behaviors. We present here a method that tries to identify and learn independent "basic" behaviors solving separate tasks the agent has to face. Using a combination of these behaviors (an action-selection algorithm), the agent is then able to efficiently deal with various complex goals in complex environments.

1. Introduction

Considering the design of autonomous agents, an important difficulty we met was to learn a policy not only in a partially observable environment (without model), but also considering various goals. Such a situation is shown by the example given on figure 1, where an agent has to push tiles in holes while avoiding the holes. A classical RL algorithm (Kaelbling et al., 1996) may only find out how to avoid the holes, what is a low local optimum.

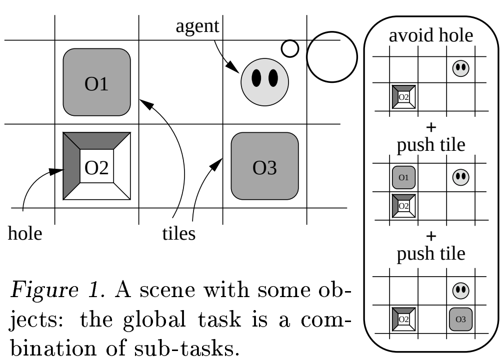

Figure 1. A scene with some objects: the global task is a combination of sub-tasks.

Tools developped to precisely answer to such problems of managing multiple goals belong to the field of Action Selection (AS) (Tyrrell, 1993). Yet, existing approaches are generally non-adaptive winner-take-all algorithms (selecting *one* behavior to apply) and produce deterministic behaviors. We have thus proposed our own algorithm (see (Buffet et al., 2002)), which has moreover made it possible to overcome some local optima (Buffet et al., 2003) and also to automatically determine the "basic behaviors" the agent requires.

2. Basic Behaviors' Combination

We in fact studied several Action Selection algorithms, all based on simple combinations of *stochastic* policies. As we work in an "object-oriented" environment, such a policy is defined for a set of types of objects. Additionally, the knowledge of the corresponding Q-values has proved to be useful to balance the policies (a policy gives an opinion with a strength depending on the Q-values). From these remarks, we can go to the definition of a **behavior** b as a tuple $\langle \mathcal{C}_b^T, P_b, Q_b \rangle$, where:

1. $\mathcal{C}_b^T$ is a **type of configuration**, that is, a tuple of types of objects involved in the behavior.
2. $P_b(o, c, a)$ is a **stochastic decision policy**. Given an appropriate configuration, it maps its observations to probability distributions over actions.
3. $Q_b(o, c, a)$ is the **Q-table of this policy**, giving the expected discounted reward of an observation.

Having selected a set $\mathcal{B}$ of "basic" behaviors (bbs) (as `[avoid hole]` and `[push tile]` in the case of Figure 1), the action selection mechanism has to identify the subsets of perceived objects (called *configurations*) corresponding to a bb's type of configuration. Having for each $b \in \mathcal{B}$ the set of current configurations $\mathcal{C}(b, o)$, the probability of choosing action a under observation o is obtained by (we give only an example of a satisfying computation we experimented):

$$Pr(a|o) = \frac{1}{K} \cdot \frac{1}{k_{(o,a)}} \sum_{b \in \mathcal{B}} \underbrace{e^{\theta_b}}_{to\ learn} \underbrace{\Big[\sum_{c \in \mathcal{C}(b,o)} |Q_b(o,c,a)|.P_b(o,c,a) \Big]}_{already\ known}$$

$$(\text{ with } k_{(o,a)} = \sum_{(b,c) \in \mathcal{BC}(o)} e^{\theta_b}.|Q_b(o,c,a)| \ ).$$

In these formulas appears a set of θ_b parameters used to automatically balance the *bb*s, which may have different relative importances from one problem faced by the agent to another. The adaptation has simply been made through simulated annealing.

3. Incremental Learning

The results of the combination described above are globally satisfying: the policy observed is near an optimal one in most cases. Yet, some particular situations are problematic, since there "non-linear" nature makes it impossible for the combination to propose an appropriate decision (see Fig. 2).

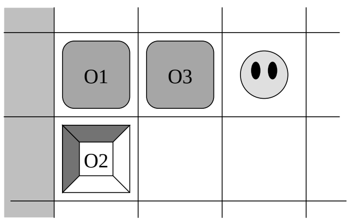

Figure 2. Blocking with 2 tiles and 1 hole (and an east wall): Both "push" basic behaviors suggest to go west, whereas the right option is to go north.

This problem encouraged us to use this policy obtained by the combination as an *initialisation* for a direct policy search (through a gradient ascent (Baxter et al., 2001)). This process led to really good stochastic policies overcoming local optima in which classical approaches would have felt. The problem is then to decide if this new policy must be added to the existing *bb*s...

4. A Growing Tree of BBs

From this point, it was natural to think of a method to automatically determine the set $\mathcal{B}$ of the agent's basic behaviors. This just requires:

- exploring a tree of behaviors to evaluate, this tree being ordered according to: 1- growing types of configurations and 2- growing combinations of rewards (in our example, the elementary rewards correspond to the two intuitive behaviors used in Fig. 1), and
- deciding if a new behavior b_n must be retained in $\mathcal{B}$ thanks to a chosen criterion which measures the knowledge it brings.

The result is shown on Fig. 3, where both intuitive *bb*s have been found, along with the *bb* resolving Fig. 2's blocking.

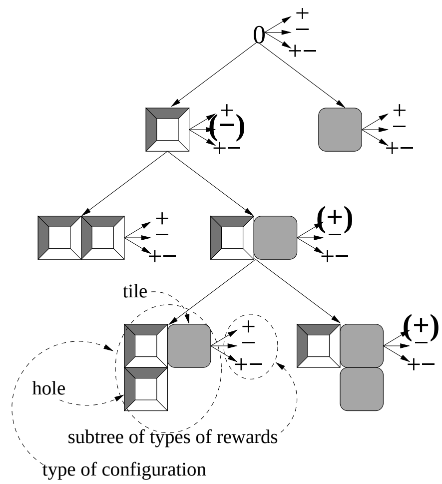

Figure 3. The tree of tested behaviors and (in parenthesis) the ones kept (the '+' and '-' signs indicate the use of the positive or negative elementary rewards).

5. Conclusion

The scheme proposed here is a simple heuristic and suffers from the approximations it is based on. Yet, the agent's autonomy is important –as it is able to find its required *bb*s, what is rarely done– and this process is an interesting *progressive* approach for the design of intelligent entities.

References

Baxter, J., Bartlett, P., & Weaver, L. (2001). Experiments with infinite-horizon, policy-gradient estimation. *Journal of Artificial Intelligence Research*, *15*, 351–381.

Buffet, O., Dutech, A., & Charpillet, F. (2002). Adaptive combination of behaviors in an agent. *Proc. of ECAI'02*.

Buffet, O., Dutech, A., & Charpillet, F. (2003). Automatic generation of an agent's basic behaviors. *Proc. of AAMAS'03*. [to appear].

Kaelbling, L., Littman, M., & Moore, A. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, *4*, 237–285.

Tyrrell, T. (1993). *Computational mechanisms for action selection*. Doctoral dissertation, University of Edinburgh.