Emergence of Memory in Neuroevolution: Impact of Selection Pressures

Charles Ollion
ollion@isir.upmc.fr

Tony Pinville
pinville@isir.upmc.fr

Stéphane Doncieux
doncieux@isir.upmc.fr

ISIR, Université Pierre et Marie Curie-Paris 6, CNRS UMR 7222
4 place Jussieu, F-75252, Paris Cedex 05, France

ABSTRACT

How to drive a learning process towards the emergence of a memory? It is hypothesized here that a reward function which evaluates the fulfillment of a task requiring memory does not necessarily reward the stepping stones to this cognitive ability. This question is studied from an evolutionary robotics perspective. Both structure and parameters of a neural network supposed to exhibit a memory are generated through an evolutionary search. Results show that selective pressures driving the evolutionary search are of critical importance. We further hypothesize that one feature of controllers with a memory is their ability to exhibit consistent behaviors over different contexts. To validate this hypothesis, a new fitness objective rewarding behavior consistency in different contexts is introduced and tested on a T-maze ER task — a task involving both navigation and working memory. The efficiency of the fitness objective is studied, as well as its effects on the overall performance and generalization ability of the controller. Results show that it is complementary to a behavioral diversity objective, thus leading to improved results when using both selection pressures.

Categories and Subject Descriptors: I.2.6 [Artificial intelligence]: Learning

General Terms: Algorithms.

Keywords: Evolutionary Algorithm, Evolutionary Robotics, Neuroevolution, Selection pressure.

1. INTRODUCTION

To exhibit a behavior that needs to remember past events, a robot requires a dedicated controller architecture that can store information. Learning over such architectures is a difficult task. Evolutionary algorithms are versatile optimization algorithms that can be used to find an adapted architecture and/or its corresponding parameters. Neural networks in particular can exhibit such properties and be optimized with evolutionary algorithms. Following the seminal work of Yamauchi and Beer [Yamauchi and Beer, 1994], most works on this topic have focused on network structures [Ziemke, 1999, Capi and Doya, 2005], but generating such networks with evolutionary algorithms still remains a challenge [Blynel and Floreano, 2003]. Evolutionary search proceeds by balancing diversification that consists in exploring the search space with intensification that consists in optimizing the best solutions found so far. These two different aspects of EA result from the exploration done by the genetic operators (mutation and, eventually cross-over) together with the selection algorithm that relies on fitness values. We will refer to the fitness function and all mechanisms influencing the selection process as *selection pressures*. In this work, we will hypothesize that the difficulty to generate neural networks with a memory is not (at least not only) a problem of network structure or encoding, but rather a problem of selection pressure. The question we will address is then: *what selection pressure should we use to drive the evolutionary search towards controllers exhibiting memory?*

A selection pressure should drive the evolutionary search from randomly generated individuals to desired solutions. We hypothesize here that evolving a memory is a deceptive task, i.e. that intuitive goal oriented fitness functions are misleading. More precisely, we think that reactive controllers represent a very attractive local optima that is difficult to escape from and the contribution of this work aims at enhancing both diversification and intensification phases to solve this problem. The first contribution consists in showing the impact of behavioral diversity [Mouret and Doncieux, 2012] for the evolution of memory, while it has been tested mostly on reactive controllers up to now. Behavioral diversity is a selection pressure that is independent from memory and aims at enhancing the diversification part of the evolutionary algorithm. The second contribution is the proposition of a new selection pressure dedicated to the emergence of an internal representation. This selection pressure explicitly rewards networks that exhibit some form of memory. It has been designed with the goal to be compatible with any kind of neural network encoding and without making any assumption on where the memory should emerge. These two contributions have been tested on a T-maze navigation task requiring to memorize some inputs to generate the expected behavior.

2. METHODS

In the following, two helper objectives have been considered in a multi-objective scheme:

- a behavioral diversity, as defined in [Mouret and Doncieux, 2012];
- a consistency objective, as introduced in this work.

The behavioral diversity assumes a distance $d_b(x, y)$ between the behaviors x and y in a population of N individuals. The diversity associated with individual x is then computed

GECCO'12 Companion, July 7–11, 2012, Philadelphia, PA, USA.
ACM 978-1-4503-1178-6/12/07.

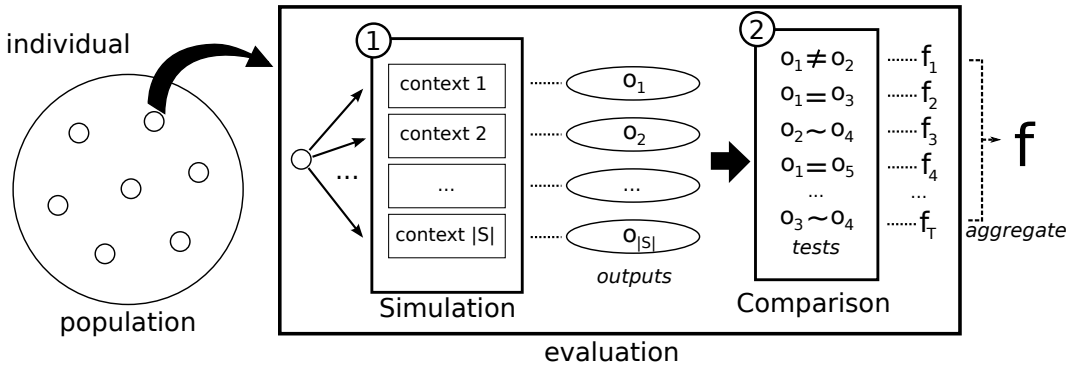

Figure 1: Details of the evaluation of an individual by the consistency objective. 1) An individual (here a neural network with internal neurons $n_1, n_2, ...$) is simulated onto several predefined contexts. During this simulation, the behavior of each internal neuron is stored. 2) The internal behaviors are compared and checked for coherence, resulting in a partial fitness value f_i. Then, the partial fitness values are aggregated into the final evaluation f.

in the following way:

$$div(x) = \frac{1}{N-1} \sum_{y \neq x} d_b(x, y)$$

The behavioral distance d_b is specific to each experiment.

The generic framework for the consistency objective is described in Figure 1. An individual is simulated over a collection of predefined contexts. Its behavior on the different contexts is stored (here the behavior of internal neurons is considered). The fitness value of the objective is derived from the comparison of those behaviors.

The definition of contexts and their comparisons depends on the considered task, and will thus be described in the next section.

3. EXPERIMENTAL SETUP

3.1 T-Maze navigation task

The task is an extension of the "roadsign problem" [Ziemke and Thieme, 2002]: an agent starts off at the bottom of a T-shaped maze, encounters an instruction stimulus (e.g. a light) while moving along a corridor and, when it reaches the junction, it has to turn left or right, depending on which stimulus has been encountered (Figure 2).

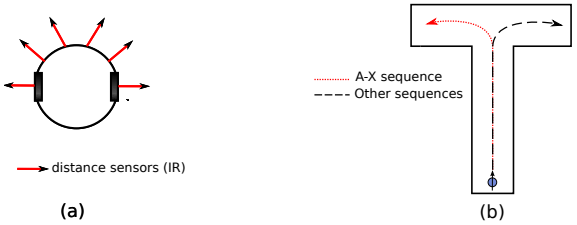

Figure 2: (a) Simulated mobile robot used for the T-maze task. The robot has four additional sensors, one by letter. (b) Map employed for this task.

To make the task more cognitive, the instruction stimulus is a combination of four stimuli (A, B, X, Y) following the same rule as in the AX-CPT working memory test [Braver et al., 1995, Pinville and Doncieux, 2010]. This task consists of a context cue (A or B), followed by a probe (X or Y) after some delay. The agent must turn to the left when the stimulus A is followed by the stimulus X, and to the right otherwise (for AY, BX, BY).

The agent is a simulated two-wheeled robot receiving sensory inputs from six infrared distance sensors and four letter sensors, one sensor for each letter A, B, X, Y, which receives 1 if the letter is presented, 0 otherwise. The robot controls its speed through two output units corresponding to its left and right motors. The agent is evaluated on each letter sequence (A followed by X, AY, BX, BY). The fitness increases by one if it turns to the correct side for the sequences AY, BX, BY and by three for the sequence AX, for a maximal value of 6. This fitness (normalized) will be referred to as "Goal oriented fitness".

Both motors are disabled during the presentation of the letters. The whole task lasts 350 steps and takes place as follows with t the number of elapsed time steps:

- $0 < t < 50$: presentation of the first letter (A/B);
- $50 \leq t < 100$: delay, all the sensors are set to 0;
- $100 \leq t < 150$: presentation of the second letter (X/Y);
- $150 \leq t \leq 350$: the robot can move and must reach the correct side of the T-maze.

In order to avoid overfitting to a specific initial configuration of the robot, 12 different setups have been defined for each possible letter sequence. A setup is described by an initial starting position (4 different positions) and an initial starting angle (3 different angles).

3.2 Neural network encoding

The agent is controlled by a neural network whose structure and parameters are evolved. DNN, a simple direct encoding has been used [Mouret and Doncieux, 2009b, Mouret and Doncieux, 2009a]. It does not use crossover. Mutations

can change parameters (connection weights and neuron biases) and add or remove neurons or connections. A lPDS-based (locally Projected Dynamic System) neuron model [Girard et al., 2008] is used to simulate the neurons with an output in $[-1, 1]$. It corresponds to a variant of the leaky integrator model with similar dynamics but with the dynamic property of contraction [Girard et al., 2008]. The same setup has already been used in [Pinville et al., 2011].

3.3 Consistency Objective

The Consistency objective evaluates controlleurs for all 12 different setups. For each setup, the 4 letter sequences define four contexts. For each controller, the behavior (output) of each internal neuron is stored. As the computation of the goal-oriented fitness already requires the simulation of the robot behavior on these contexts, no additional evaluation is required.

An individual has N internal neurons — N may vary from individuals to individuals and during evolution. $b^i_s(t)$ is the output of the i-th internal neuron in context s at time-step t, after the presentation of letters ($t > 150$). The goal of the consistency objective is to force individuals to obey the following rules:

$$\forall s \in S,\ b^i_{AX}(t) \neq b^i_s(t)$$

$$\forall s, s' \in S,\ b^i_s(t) = b^i_{s'}(t)$$

where $S = \{AY, BX, BY\}$. In other words, the consistency objective rewards the existence of at least one internal neuron that exhibits a similar behavior for AY, BX, BY contexts, and a different one for AX contexts. The behavior is computed after the presentation of letters, i.e. when the input letters are no longer active. The existence of a difference between the contexts should reflect the emergence of a memory.

For each internal neuron i two partial fitnesses f^i_1 and f^i_2 are computed, they measure how well the internal neuron respects the two previous rules:

$$f^i_1 = \frac{1}{|S|} \sum_{s \in S} \frac{1}{T} \sum_{t}^{T} \frac{|b^i_{AX}(t) - b^i_s(t)|}{2}$$

$$f^i_2 = 1 - \Big[\frac{1}{|S|^2 - |S|} \sum_{s2 \in S} \sum_{s \neq s2} \frac{1}{T} \sum_{t}^{T} \frac{|b^i_s(t) - b^i_{s2}(t)}{2} | \Big]$$

The fitness of each internal neuron is computed as follows:

$$f^i = f^i_1 + f^i_2$$

As the goal of this experiment is to select individuals that have *at least* one internal neuron that represents the information, the final fitness is computed as the maximum of all internal fitnesses f^i:

$$f = \max_{0 \leq i < N} f^i$$

The fitnesses f compare the four letter sequences evaluated in the same setup. The overall consistency objective corresponds to the average of the 12 fitnesses thus defined (one for each setup).

To test the influence of each objective, experiments are launched with various combinations of objectives:

	Setup	Objectives to be optimized
1	G	Goal-oriented
2	G + D	Goal-oriented + Diversity
3	G + C	Goal-oriented + Consistency
4	G + D + C	Goal-oriented + Div. + Consistency

The multi-objective evolutionary algorithm is NSGA-II and each of the setups is run 30 times.

4. RESULTS

Figure 3 shows that a simple fitness rewarding the completion of the task has poor results. This is confirmed by Figure 4 in which one can see that a fitness plateau is quickly reached. The fitness plateau is at $f = 0.5$, which corresponds to controllers that always go to the same side of the maze. Adding a diversity objective significantly increases performance and delays fitness plateaus. This result is compatible with our hypothesis that the evolution of a memory is a deceptive problem and shows that selective pressures have indeed a significant impact on the success rate.

The use of the Consistency Objective also increases the performance significantly, to the same extent as the diversity objective. There is no statistical difference between $G + D$ and $G + C$ setups.

Using both objectives further increases performance, and as no fitness plateau was reached during the 2000 generations (Figure 4). One can then expect the fitness to be even better with more generations.

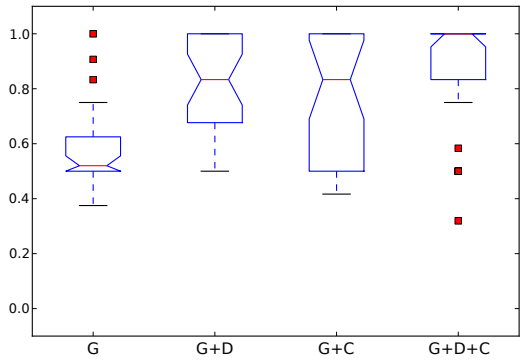

Figure 3: Best of run values for the goal-oriented objective (30 runs for each setup).

A network is considered to exhibit a reliable memory if at least one internal neuron respects the two following points: (1) After presentation of the letters, the neuron has a different output for AX contexts and for AY, BX, BY contexts. (2) The memory is not affected by the duration of the presentation of the letters. While during evolution the duration of the presentation was 50 time-steps for each letter, the activity of the network is tested —after evolutionary process— with a duration of 400 time-steps. This is aimed to detect networks that rely on complex dynamics to have different activities after exactly 50 time-steps, but would not work with a different duration. In the same way, we assess the generalization ability by testing an individual on 180 setups unseen during evaluation.

While diversity objective slightly encourages memory, the consistency objective significantly affects memory emergence (figure 5). Interestingly using both helper objectives at the

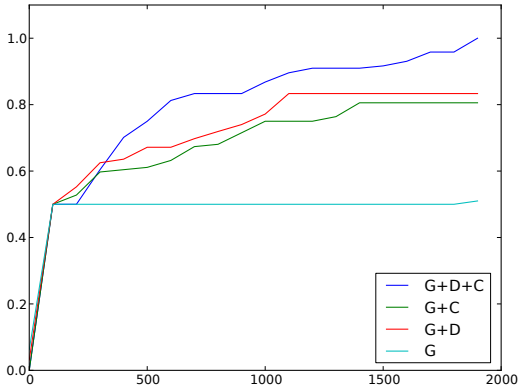

Figure 4: Evolution of fitness objective (median value of all 30 runs).

same time has less impact on memory than the consistency objective alone. Figure 5 also shows that the diversity and consistency objective significantly increase the generalization ability.

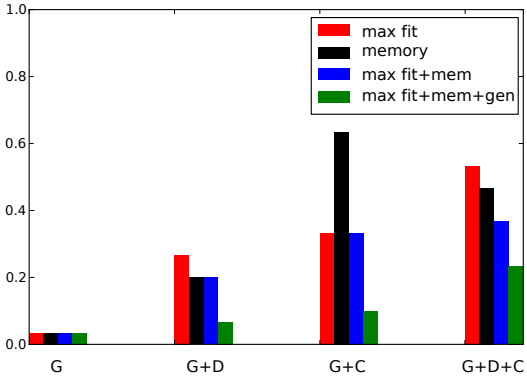

Figure 5: Proportion of runs matching different criteria: (1) achieving maximal fitness (2) having memory (3) having both (4) having both and generalizing to 60 of the 180 extra setups.

5. CONCLUSION

These experiments confirm that the emergence of memory is a challenging problem. With the simple direct encoding used in these experiments, structures with memory require several mutations to appear. They are thus unlikely to appear without paying particular attention to selective pressure. The helper objectives considered, both diversity and the newly defined consistency objective, significantly increase the convergence rate on this task.

The consistency objective —and, to a lesser extent, the diversity objective— promote memory in the resulting networks. Moreover, the helper objectives are shown to have a large impact on generalization ability even though they aren't specifically designed to do so. We can hypothesize that there is a link between the presence of memory in agents and the generalization ability on this task. The consistency objective does not assume a specific structure and could potentially be used in any neuroevolution experiment.

Another methodological aspect highlighted in this paper is the use of a multi-objective evolutionary algorithm. New objectives are simply added to reward individuals that have a low goal-oriented fitness value, but an original behavior or an efficient internal representation. No new parameter is introduced concerning the relative weight of these different objectives.

6. ACKNOWLEDGMENTS

This project was funded by the ANR EvoNeuro project, ANR-09-EMER-005-01.

7. REFERENCES

[Blynel and Floreano, 2003] Blynel, J. and Floreano, D. (2003). Exploring the T-maze: Evolving learning-like robot behaviors using CTRNNs. *Applications of Evolutionary Computing*, pages 593–604.

[Braver et al., 1995] Braver, T. S., Cohen, J. D., and Servan-Schreiber, D. (1995). A computational model of prefrontal cortex function. *Nips*, pages 141–148.

[Capi and Doya, 2005] Capi, G. and Doya, K. (2005). Evolution of Neural Architecture Fitting Environmental Dynamics. *Adaptive Behavior*, 13(1):53–66.

[Girard et al., 2008] Girard, B., Tabareau, N., Pham, Q. C., Berthoz, A., and Slotine, J. J. (2008). Where neuroscience and dynamic system theory meet autonomous robotics: a contracting basal ganglia model for action selection. *Neural Networks*, 21(4):628–641.

[Mouret and Doncieux, 2009a] Mouret, J.-B. and Doncieux, S. (2009a). Overcoming the bootstrap problem in evolutionary robotics using behavioral diversity. In *IEEE Congress on Evolutionary Computation, 2009 (CEC 2009)*, pages 1161–1168.

[Mouret and Doncieux, 2009b] Mouret, J.-B. and Doncieux, S. (2009b). Using Behavioral Exploration Objectives to Solve Deceptive Problems in Neuro-evolution. In *GECCO'09: Proceedings of the 11th annual conference on Genetic and evolutionary computation*, pages 627–634. ACM.

[Mouret and Doncieux, 2012] Mouret, J.-B. and Doncieux, S. (2012). Encouraging Behavioral Diversity in Evolutionary Robotics: An Empirical Study. *Evolutionary computation*, 20(1):91–133.

[Pinville and Doncieux, 2010] Pinville, T. and Doncieux, S. (2010). Automatic Synthesis of Working Memory Neural Networks with Neuroevolution Methods. In *Cinquième conférence française de Neurosciences Computationnelles (Neurocomp'10)*.

[Pinville et al., 2011] Pinville, T., Koos, S., Mouret, J.-B., and Doncieux, S. (2011). How to Promote Generalisation in Evolutionary Robotics : the ProGAb Approach Formalising the Generalisation Ability. In *GECCO '11: Proceedings of the 13 th annual conference on Genetic and Evolutionary Computation*, pages 259–266.

[Yamauchi and Beer, 1994] Yamauchi, B. M. and Beer, R. D. (1994). Sequential Behavior and Learning in Evolved Dynamical Neural Networks. *Adaptive Behavior*, 2(3):219.

[Ziemke, 1999] Ziemke, T. (1999). Remembering how to behave: Recurrent neural networks for adaptive robot behavior. *Recurrent neural networks: Design and applications*, pages 341–376.

[Ziemke and Thieme, 2002] Ziemke, T. and Thieme, M. (2002). Neuromodulation of reactive sensorimotor mappings as a short-term memory mechanism in delayed response tasks. *Adaptive Behavior*, 10(3/4):185–199.