With a little help from selection pressures: evolution of memory in robot controllers

Charles Ollion, Tony Pinville, Stéphane Doncieux

With a little help from selection pressures: evolution of memory in robot controllers

Ollion Charles[1,2], Pinville Tony[1,2] and Doncieux Stéphane[1,2]

[1]- UPMC Univ Paris 06, UMR 7222, ISIR, F-75005, Paris, France
[2]- CNRS, UMR 7222, ISIR, F-75005, Paris, France
ollion,pinville,doncieux@isir.upmc.fr

Abstract

Evolutionary robotics (ER) have successfully built robot controllers presenting a reactive behavior. However, the evolution of cognitive controllers is still a challenge. We hypothesize here that a fitness function which rewards the fulfillment of a task requiring cognitive abilities does not necessarily reward the stepping stones that lead to cognitive controllers. In other words, our hypothesis is that evolving cognitive abilities is a deceptive problem; so that the selective pressures driving the evolutionary search are of critical importance. This paper presents some experiments to confirm this hypothesis and addresses this selective pressure problem by introducing a new helper-objective that rewards controllers with a memory. This is potentially useful for the design of controllers in which an internal representation of some data is required to solve a task. It does not assume how the memory is stored in the controller, therefore reducing the bias towards a particular solution. The new objective is tested in a multi-objective scheme on a T-maze ER task — a task involving both navigation and working memory. The efficiency of the helper-objective is studied, as well as its effects on the overall performance and generalization ability of the controller.

Introduction

Evolutionary Robotics deals with the use of evolutionary algorithms (EA) in the design process of robots (Doncieux et al., 2011; Floreano et al., 2008b). Such algorithms have been used for various tasks (Nelson et al., 2009). Typical studies deal with the evolution of locomotion controllers (Allen and Faloutsos, 2009) or obstacle avoidance controllers, (Durand et al., 2000), that are mostly reactive behaviors, i.e. behaviors that usually do not require any memory of past actions or perceptions.

Cognition may refer to a wide range of abilities: from capacities specific to humans (Dennett, 1997) to any ability of a living organism (Maturana and Varela, 1980; Heschl, 1990). Here we will use it to describe abilities that go beyond reactive behaviors and require to take past actions and/or perceptions into account while choosing how to move and what to do. A neural network can exhibit such abilities thanks to a recurrent network structure or to some form of plasticity. The question we will address here is the evolution of such cognitive abilities with a focus on non-plastic recurrent networks.

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, 2005a). One problem remains when evolving such systems: the evolvability. Evolving such networks is a challenge (Blynel and Floreano, 2003) and we may wonder why. Evolutionary search proceeds by balancing diversification, which consists of exploring the search space, with intensification, which consists of optimizing the best solutions found so far. These two different aspects of EA result from the exploration done by the genetic operators (mutation and 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 of generating cognitive abilities 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 to use to drive the evolutionary search towards controllers with cognitive abilities?*

A selection pressure should drive the evolutionary search from randomly generated individuals to desired solutions. We hypothesize here that evolving cognitive abilities 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 cognitive abilities, while it has been tested mostly on reactive controllers up to now. Behavioral diversity is a selection pressure that is independent from the cognitive abilities we are looking for. 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

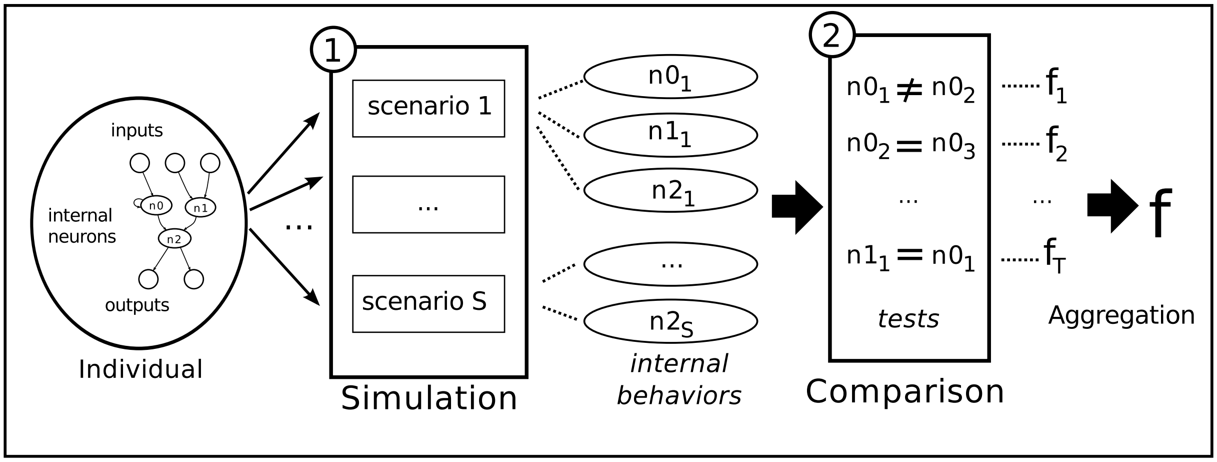

Figure 1: Details of the evaluation of an individual by the scenario-based objective. 1) An individual (here a neural network with internal neurons $n_1, n_2, ...$) is simulated in several predefined scenarios. During this simulation, the behavior of each internal neuron is stored. 2) The internal behaviors 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.

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 a priori 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.

Related Work

Evolution of cognitive abilities

Two main kinds of cognitive abilities have been investigated so far: memory (Ziemke, 1999; Ziemke and Thieme, 2002) and learning (Blynel and Floreano, 2003; Floreano and Urzelai, 2001). This work proved that it was possible to generate such capabilities with an evolutionary approach, but it still remains a challenge. Such contributions can be roughly divided in two different categories (Floreano et al., 2008a). Following the seminal work of Yamauchi and Beer (1994), the first category studies continuous time recurrent neural networks without plasticity. Multiple experiments have thus shown that, through evolutionary optimization, such networks can exhibit a memory capability (Ziemke, 1999; Blynel and Floreano, 2003; Capi and Doya, 2005a). The second category focuses on learning and neuromodulation and tries to evolve networks with plastic connections (Floreano and Urzelai, 2000; Ziemke and Thieme, 2002; Tonelli and Mouret, 2011). Both kinds of work mainly focus on the features of the neural network structure (completely connected neural network, Elman network or others) or on the encodings that allows to explore network structures with evolutionary algorithms.

Selection pressures

Floreano and Urzelai (2000) proposed a framework for describing fitness functions: *the fitness space*. Recognizing the importance of the fitness function definition on the results of an ER experiment, they proposed a classification of fitness functions in order to easily allow their qualitative description, assessment and comparison. Nelson et al. (2009) have made a review of the different fitness functions used in ER classified according to the degree of a priori knowledge incorporated in the fitness. Both works recognized the impact on performance of the fitness function, but none of them aimed at better understanding it.

Lehman and Stanley (2008, 2011) have shown how deceptive goal-oriented fitness functions can be. The novelty search approach they have proposed consists in looking for novel solutions instead of efficient ones. Associated with the increasing complexity feature of the NEAT encoding (Stanley and Miikkulainen, 2002), they have shown that, on different problems, such an exploration was much more efficient than a search driven by a distance towards a goal to be reached. This counterintuitive result has shown how strong the impact of the selection pressure is.

Several studies did propose to take into account a space that is specific to ER, i.e. the behavioral space, in the diversification phase. Trujillo et al. (2011) proposed a speciation mechanism based on behavior, while Gomez (2009); Mouret and Doncieux (2009b,a) proposed to use behavioral distances for diversity preservation. Mouret and Doncieux (2012) made several comparisons with the following conclusions: (1) explicitly encouraging behavioral diversity leads to substantial improvements (2) multi-objective approaches lead to better results.

The impact of selection pressure on the evolution of cognitive abilities has been seldom studied. Capi and Doya (2005b) have shown that an evolutionary algorithm inspired from island models facilitates the evolution of memory, thus suggesting that the selection pressure has an impact on cognitive ability evolution. The goal of this paper is first to confirm the importance of selection pressure to the evolution of memory, and to propose fitness functions that promote this evolution.

Methods

The multi-objective approach has an interesting feature: adding a selection pressure can be done simply by adding as a separate objective with no need to tune any new parameter for the relative importance of each objective to be optimized. This means that all objectives are considered equally important and multi-objective evolutionary algorithms aim at finding the best trade-off solutions relative to them (Deb, 2001). The two selection pressures studied here are then defined as separate objectives to be optimized with a multi-objective evolutionary algorithm. Such objectives do not describe the goal to be reached, but aim at enhancing the evolutionary search, they are then *helper objectives*. This approach is called multiobjectivization (Knowles et al., 2001; Mouret, 2011).

In the following, two helper objectives have been considered:

- a behavioral diversity, as defined in Mouret and Doncieux (2012);
- a scenario-based objective, as introduced in this work.

Behavioral diversity

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

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

The behavioral distance d_b will be described in the Experimental Setup Section.

Scenario-based objective

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

Scenario-based objectives promote individuals with a consistent behavior without explicitly describing a target behavior. For instance in order to promote robustness to noise, individuals could be simulated on scenarios with various levels of noise. Individuals that have close behaviors in those scenarios should be rewarded, while individuals whose behaviors are strongly affected by noise should be punished.

Behaviors of an individual are compared and the scenario based objective will reward either their similarity or difference. The design of this objective actually consists in defining the scenarios and choosing whether the corresponding behavior should be similar or different one with another. By rewarding the similarity between behaviors or, in contrast, their difference, the scenario based objective encourages the emergence of a coherent behavior.

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

Experimental Setup

T-Maze navigation task

The task is an extension of the “roadsign problem” (Ziemke and Thieme, 2002; Rylatt and Czarnecki, 2000): 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).

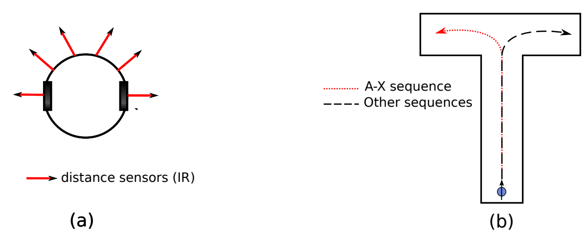

Figure 2: (a) Simulated mobile robot used for the T-maze task. The robot has four additional sensors, one for each letter. (b) Map employed for this task.

In the initial setup, controllers that simply follow the right or left wall after the signal can solve the task while not having any memory (Ziemke and Thieme, 2002). To make this task more cognitive, in our experiment 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 stimulus (A or B), followed by a second stimulus (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).

Here, the agent is a simulated two-wheeled robot receiving sensory inputs from 6 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 1 if it turns to the correct side for the sequences AY, BX, BY and by 3 for the sequence AX, for a maximal value of 6. This fitness will be referred 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 contexts have been defined for each possible letter sequence. A context is described by an initial starting position (4 different positions) and an initial starting angle (3 different angles).

The behavioral distance d_b between two individuals used to compute the behavioral diversity is the euclidian distance between the positions of the two robots at $t = 350$.

Neural network encoding

The agent is controlled by a neural network whose structure and parameters are evolved. DNN, a simple direct encoding inspired from NEAT (Stanley and Miikkulainen, 2002) has been used (Mouret and Doncieux, 2009b,a). 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 classic leaky integrator 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).

Scenario-based Objective

Each individual is simulated and evaluated on the 12 different contexts. In each of these contexts, one individual is simulated over the 4 different scenarios AX, BX, AY, and BY.

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

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

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

With $S = \{BX, AY, BY\}$. In other words, the behavior of an internal representation should be the same if the inputs are AY, BX, BY, and different if the input is AX. The behavior is computed after the presentation of letters, which means that the input letters are no longer active. The existence of a difference between the scenarios should reflect the emergence of a memory.

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

$$f^i_1 = \frac{1}{|S|} \sum_{s \in S} \frac{1}{200} \sum_{t=150}^{350} \frac{|b^i_{AX}(t) - b^i_s(t)|}{2}$$

$$f^i_2 = 1 - \Big[\frac{1}{|S|^2 - |S|} \sum_{s,s' \in S, s \neq s'} \frac{1}{200} \sum_{t=150}^{350} \frac{|b^i_s(t) - b^i_{s'}(t)|}{2} \Big]$$

Then, 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 context. The overall scenario-based fitness corresponds to the average of the 12 fitnesses thus defined (one for each context).

Setups summary

Throughout the article, we will refer to the different objectives as follows:

- **G**: Goal-oriented objective;
- **D**: Diversity objective;
- **S**: Scenario-based objective;

To test the influence of each objective, experiments are launched with various combination of objectives as shown in Table 1. The multi-objective evolutionary algorithm is NSGA-II (Deb, 2001) and each of these setups is run 30 times.

Results

Figure 3 depicts boxplots for the goal-oriented fitness results on each different setups. The red line represents the median value, the box extends from the lower to upper quartile values of the data. The whiskers extend from the box to show

Table 1: Summary of different setups used

	Setup	Description
1	G	Goal-oriented
2	G + D	Goal-oriented + Diversity
3	G + S	Goal-oriented + Scenario-based
4	G + D + S	Goal-oriented + Diversity + Scenario-based

the range of the data. Flier points are those past the end of the whiskers.

Table 2 displays the corresponding p-values using Mann-Whitney statistical test. Figure 4 shows the median fitness values for the 4 setups.

Diversity Effect

Figure 3 shows that a simple fitness rewarding the completion of the task (G) 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 (D) 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.

Scenario-Based Objective Effect

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

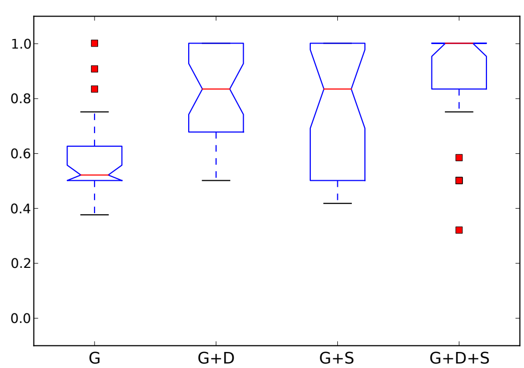

Figure 3: Boxplots for the goal-oriented fitness results on each different setups

Table 2: P-values between each setup on goal-oriented fitness value

	G + D + S	**G + S**	**G + D**	**G**
G + D + S	x	0.04013	0.0639	< 1e-05
G + S	0.04013	x	0.18504	0.00409
G + D	0.0639	0.18504	x	< 1e-05
G	< 1e-05	0.00409	< 1e-05	x

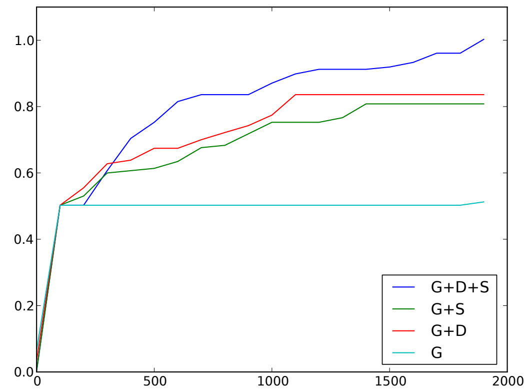

Figure 4: Evolution of fitness objective (median value of all 30 runs).

The two next sections present a more in depth study of results: the resulting networks are tested for reliable memory and generalization ability.

Memory computation A network is considered to exhibit a reliable memory if at least one internal neuron respects the two following points:

- After presentation of the letters, the neuron has a different output for AX scenarios and for AY, BX, BY scenarios.
- 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 Figure 5, the black histogram displays the percentage of runs (out of the 30 runs per setup) in which the best individual achieves reliable memory. While diversity objective slightly increases memory, the Scenario-based objective significantly affects memory. Interestingly using both helper objectives results in less memory than using the Scenario-objective alone.

Generalization ability Another important aspect studied here is the generalization ability. *During evolution*, the robot is tested in 12 different contexts for each letter sequence, and maximal fitness is achieved only if the individual manages to solve the problem in all the contexts. *After evolution*, the best controllers are tested in 180 previously unseen contexts. The 180 new contexts include different map sizes, starting positions, and starting orientations of the robot. A controller is considered to generalize well if it can still perform the task in at least 60 of these new contexts. Figure 5 shows the proportion of runs with individuals which generalize. Figure 6 details the number of context in which these individuals generalize. There is a very significant increase of generalization when using the helper objective, and even more when using both objectives. Table 3 displays the corresponding p-values.

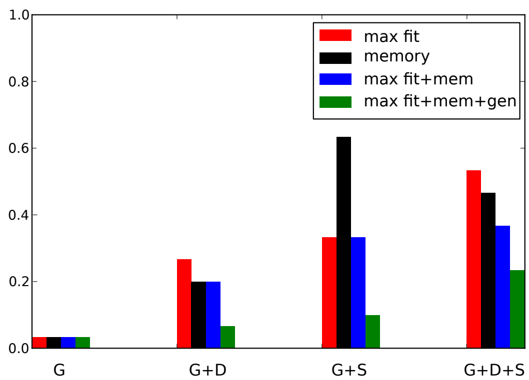

Figure 5: Proportion of runs matching different criteria: (1) achieving maximal fitness (2) having memory (3) having both (4) having both and generalizing to 60 of the 180 extra contexts.

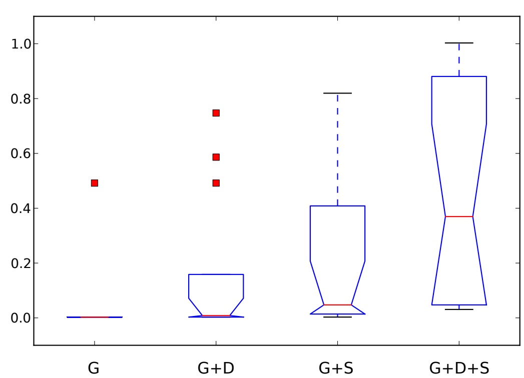

Figure 6: Generalization ability of the 15 best runs for the 4 different setups. The value corresponds to the normalized number of contexts in which the agent solves the task.

Table 3: P-values between each setup on the generalization ability

	G + D + S	**G + S**	**G + D**	**G**
G + D + S	x	0.03241	0.00364	1e-05
G + S	0.03241	x	0.08408	9e-05
G + D	0.00364	0.08408	x	0.0094
G	1e-05	9e-05	0.0094	x

Analysis of the resulting networks

Two resulting networks, shown in figures 7 and 9, are analysed in this section. They both achieve maximal fitness, but only the second one exhibits reliable memory and generalization ability. Blue neurons have a different neural activity for AX sequence than the others during the memory test. Figure 8 and 10 show the corresponding internal behavior of the neurons during the test for networks in figure 7 and 9. The first presentation of letters lasts from 0 to 400, the delay from 400 to 800, the second letter from 800 to 1200. In order to distinguish AX and BX sequences, the network must remember A or B stimulus during the delay period.

In figure 8, we can see that the network depicted on figure 7 is not able to retain A or B stimulus when the delay interval is extended. At timestep $t = 800$, the internal behavior of the neurons are similar for the 4 sequences. At the end of the presentation of letters, the neural network cannot therefore distinguish AX and BX sequences. In figure 10, there are two different neurons, neurons 0 and 3, able to memorize stimulus B even if the delay interval is extended. In this case, at the end of the presentation of letters, the internal behavior of the neurons for AX sequence is different than for the other sequences.

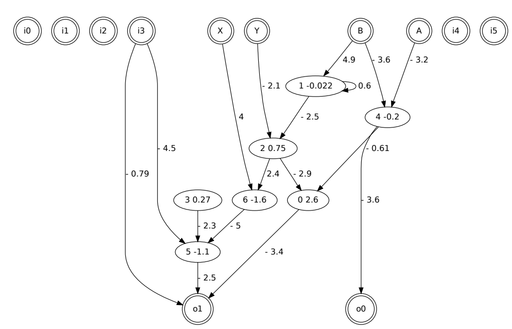

Figure 7: Resulting neural network with maximal fitness, but no memory nor generalization

Conclusion

These experiments confirm that the emergence of memory is a challenging problem. With the present encoding, structures with memory require several mutations to appear, will be much more likely to appear under specially-designed se-

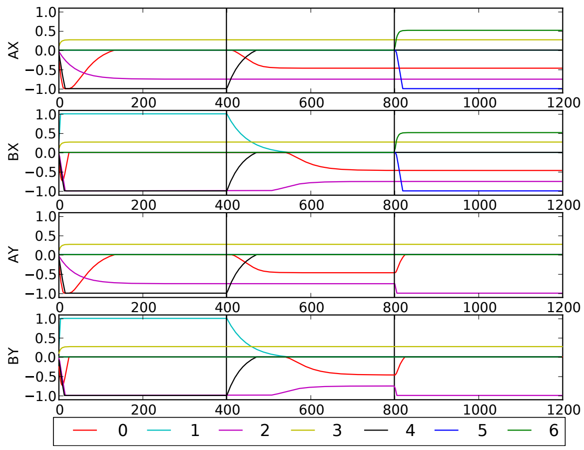

Figure 8: Internal behavior of the neurons corresponding to neural network displayed in Figure 7, for the 4 different sequences during the memory test.

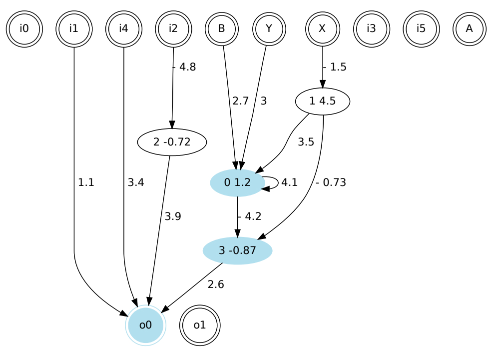

Figure 9: Resulting neural network with maximal fitness, memory and generalization

lective pressures. The helper objectives considered, both diversity and the newly defined scenario-based objective, significantly increase the convergence rate on this task.

The scenario-based 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 scenario-based method does not assume a specific structure and could potentially be used in any neuroevolution experiment involving elementary memory. The scenario-based objective is crucial here because it can select individuals with many different internal representations. Another methodological aspect highlighted in this paper is the use of a multi-objective evolutionary algorithm. Additional objectives are simply added, selecting individuals that might have a low fitness regarding to the main objective, but have an original behavior or efficient internal representation. We believe that those individuals can be good stepping stones to efficient cognitive solutions.

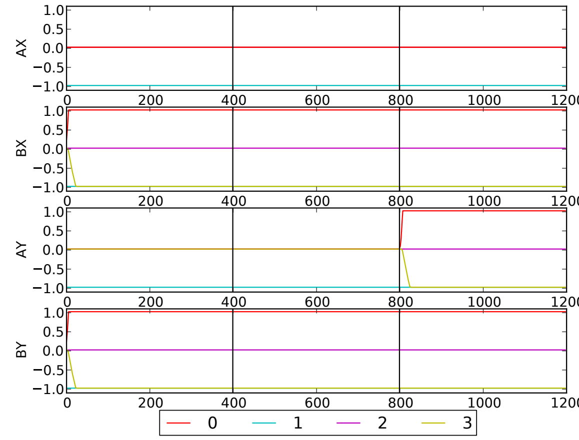

Figure 10: Behavior of internal neurons corresponding to neural network displayed in Figure 9, for the 4 different sequences during the memory test.

Future work The use of specific helper objectives and behavioral diversity objectives have a critical impact on the success rate of the presented experiments. However, Figure 5 shows room for improvement. Novelty Search (Lehman and Stanley, 2008, 2011) may also be defined as an helper objective (Mouret, 2011) and may thus be compared to the selection pressures proposed hare. It should also be noted that the scenario-based method is not specific to the task nor the encoding. It could be applied to any neuroevolution encoding, such as NEAT (provided that it is adapted to multi-objective problems), or to fixed structures such as Elman or Echo State Networks.

Acknowledgments

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

References

Allen, B. and Faloutsos, P. (2009). Complex networks of simple neurons for bipedal locomotion. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4457–4462. IEEE.

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

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

Capi, G. and Doya, K. (2005b). Evolution of recurrent neural controllers using an extended parallel genetic algorithm. *Robotics and Autonomous Systems*, 52(2-3):148–159.

Deb, K. (2001). *Multi-objectives optimization using evolutionnary algorithms*. Wiley.

Dennett, D. C. (1997). *Kinds of Minds: Towards an Understanding of Consciousness*. Basic Books.

Doncieux, S., Mouret, J.-B., Bredeche, N., and Padois, V. (2011). Evolutionary Robotics: Exploring New Horizons. In *New Horizons in Evolutionary Robotics: post-proceedings of the 2009 EvoDeRob workshop*, pages 3–25. Springer.

Durand, N., Alliot, J.-M., and Medioni, F. (2000). Neural Nets trained by genetic algorithms for collision avoidance. *Applied Artificial Intelligence*, 13(3).

Floreano, D., Dürr, P., and Mattiussi, C. (2008a). Neuroevolution: from architectures to learning. *Evolutionary Intelligence*, 1(1):47–62.

Floreano, D., Husbands, P., and Nolfi, S. (2008b). Evolutionary Robotics. In Siciliano, B. and Khatib, O., editors, *Handbook of Robotics*, pages 1423–1451, Berlin, Heidelberg. Springer Berlin Heidelberg.

Floreano, D. and Urzelai, J. (2000). Evolutionary robots with on-line self-organization and behavioral fitness. *Neural Networks*, 13(4):431–443.

Floreano, D. and Urzelai, J. (2001). Evolution of plastic control networks. *Autonomous Robots*, 11(3):311–317.

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.

Gomez, F. J. (2009). Sustaining diversity using behavioral information distance. In *Proc. of GECCO'09*, pages 113–120. ACM.

Heschl, A. (1990). L = C a simple equation with astonishing consequences. *Journal of Theoretical Biology*, 145(1):13–40.

Knowles, J. D., Watson, R. A., and Corne, D. W. (2001). Reducing Local Optima in Single-Objective Problems by Multi-objectivization. *First International Conference on Evolutionary Multi-Criterion Optimization*, 1993:268–282.

Lehman, J. and Stanley, K. O. (2008). Exploiting Open-Endedness to Solve Problems Through the Search for Novelty. In *Artificial Life*, volume 11.

Lehman, J. and Stanley, K. O. (2011). Abandoning objectives: evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223.

Maturana, H. R. and Varela, F.-J. (1980). *Autopoiesis and cognition: the realization of the living*.

Mouret, J.-B. (2011). Novelty-based multiobjectivization. In *New Horizons in Evolutionary Robotics: Extended Contributions from the 2009 EvoDeRob Workshop*, pages 139–154. Springer.

Mouret, J.-B. and Doncieux, S. (2009a). Overcoming the bootstrap problem in evolutionary robotics using behavioral diversity. In *Proc. of CEC 2009*, pages 1161–1168.

Mouret, J.-B. and Doncieux, S. (2009b). Using Behavioral Exploration Objectives to Solve Deceptive Problems in Neuro-evolution. In *Proc. of GECCO'09*, pages 627–634. ACM.

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

Nelson, A. L., Barlow, G. J., and Doitsidis, L. (2009). Fitness functions in evolutionary robotics: A survey and analysis. *Robotics and Autonomous Systems*, 57(4):345–370.

Pinville, T. and Doncieux, S. (2010). Automatic Synthesis of Working Memory Neural Networks with Neuroevolution Methods. In *Proc. of Neurocomp'10*.

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 *Proc. of GECCO '11*, pages 259–266.

Rylatt, R. M. and Czarnecki, C. A. (2000). Embedding Connectionist Autonomous Agents in Time: The 'Road Sign Problem'. *Neural Processing Letters*, 12(2):145–158.

Stanley, K. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*.

Tonelli, P. and Mouret, J.-B. (2011). On the Relationships between Synaptic Plasticity and Generative Systems. In *Proc. of GECCO '11*.

Trujillo, L., Olague, G., Lutton, E., Fernández de Vega, F., Dozal, L., and Clemente, E. (2011). Speciation in Behavioral Space for Evolutionary Robotics. *Journal of Intelligent & Robotic Systems*, 64(3):323–351.

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

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

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.

Parameters

- MOEA: NSGA-II (pop. size: 200, number of generations: 2000)
- DNN (direct encoding):
  - prob. of changing weight/bias: 0.1
  - prob. of adding/deleting a conn.: 0.15/0.25
  - prob. of changing a conn.: 0.1
  - prob. of adding/deleting a neuron: 0.025/0.025
- Source code will be available online.