



## Evolutionary Robotics: Incremental Learning of Sequential Behavior

Nicolas Bredeche, Louis Hugues

### ► To cite this version:

Nicolas Bredeche, Louis Hugues. Evolutionary Robotics: Incremental Learning of Sequential Behavior. IEEE International Conference on Development and Learning, Jul 2005, Osaka, Japan. pp.137-138. inria-00175503v2

HAL Id: inria-00175503

<https://inria.hal.science/inria-00175503v2>

Submitted on 5 Nov 2008

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Evolutionary Robotics: Incremental Learning of Sequential Behavior

Nicolas Bredeche

Inférence & Apprentissage, TAO/INRIA

Laboratoire de Recherche en Informatique

Université Paris-Sud XI - 91405 Orsay cedex – France

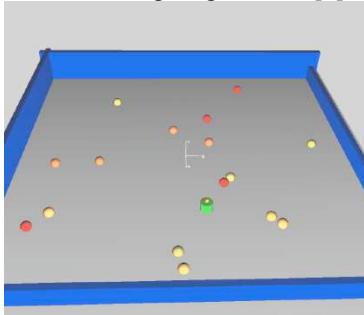
bredeche@lri.fr

**Index Terms – evolutionary robotics, dynamic fitness.**

## I. INTRODUCTION

Evolutionary Robotics offers an efficient and easy-to-use framework for automatically building behaviors for an autonomous robots. However, a major drawback of this approach relies in the difficulty to define the fitness function (i.e. *the learning setup*) in order to get satisfying results. As a consequence, many works have been limited to simple problems such as wall-avoidance or target-following [1].

On the one hand, recent works addressed this issue either by decomposing the learning task or by endowing the agent with such capabilities that should make the goal easier to achieve [1,2]. On the other hand, literature in Evolutionary Approach shows that modifying the very nature of genetic operators and/or fitness during the course of evolution may lead to better results for complex problems [3].



**Fig. 1.** The task: a robot wanders through the environment and has to perform a given behavior in order to access resources (i.e. the yellow/red balls). The behavior to perform is a sequence of “key” movements to “open” the resources. The more complex the behavior needed to access a resource, the greater the reward (one key is enough for the yellow balls while four keys are needed to “eat” the red ball). The goal is to learn the complete sequence that makes it possible to access any resources (the agent accesses simpler resources first ; performing the whole sequence gives access to any resource).

## 2. EXPERIMENTS

In the scope of this short paper, we are interested in the reformulation of a straightforward complex fitness function into more subtle versions using different approaches. The original task we are interested in is described in fig. 1<sup>\*</sup>.

Fig. 2 shows the results of our experiment. The task is successful if an individual’s genotype exactly matches a reference genotype (i.e. *the complete sequence of keys to be found*). A genotype is defined as  $N$  4-uplets of Boolean

<sup>\*</sup> A robot performs a basic random wandering behavior and automatically performs its access sequence anytime a food item is found. In the following, results are concerned only with the learning of this access sequence.

Louis Hugues

Miriad/LIP6

Laboratoire d’Informatique de Paris 6

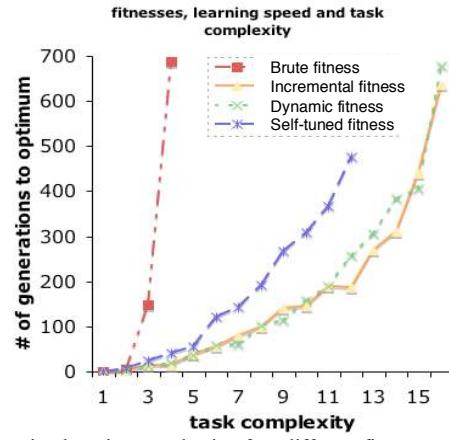
Université Paris VI - 4, place Jussieu - 75005 Paris – France

louis.hugues@wanadoo.fr

attributes. Every 4-uplets is a ‘key’ of the sequence and  $N$  is the length of the sequence (i.e. *the task complexity*). A sequence is evaluated *from the first key, key by key, until a key is found incorrect or the end is reached*. We defined four fitnesses as follow:

1. *brute fitness* : reward 1 if exact match, otherwise 0 (i.e. only the most difficult to access food items can be found in the environment);
2. *static incremental* : each accepted key is rewarded;
3. *dynamic elitism* : reward only the best individual(s);
4. *self-tuned* : each individual accepts (or not) reward for a given key (evaluation from start until error).

On the one hand, the brute fitness unsurprisingly leads to a greater search space. On the other hand, the self-tuned fitness does not perform that well – looking at the genotypes, one can see that it actually behaves like a static incremental fitness with tuning that is unfortunately disrupted by the genetic operators (there is no bias toward a relevant self-tuning). Regarding the static incremental and dynamic fitnesses, there is no real difference as the two of them rely on the selective pressure of the genetic algorithm even though dynamic elitism should be more aggressive regarding this very pressure.



**Fig. 2.** Comparing learning speed using four different fitnesses used on a more or less difficult task (i.e. sequence length up to 16 keys (i.e. 64 bits)). The x-axis shows the sequence length the agent has to learn in order to be able to get the best reward for this task (genotype length is fixed according to the sequence length). The y-axis shows the number of generations before the optimal behavior is learnt. Results with more than 1000 gen. are not shown.

## REFERENCES

- [1] Nolfi, S. and Floreano, D. "Evolutionary Robotics". MIT Press. 2001.
- [2] Godzik N., M. Schoenauer and M. Sebag: Evolving Symbolic Controllers. EvoWorkshops, pp.638-650. 2003.
- [3] Lohn J. D. et al. A Comparison of Dynamic Fitness Schedules for Evolutionary Design of Amplifiers. Evolvable Hardware, pp.87-92, 1999.