

Behavior-based Speciation for Evolutionary Robotics

Leonardo Trujillo,
Gustavo Olague
EvoVisión Project,
CICESE Research Center,
Ensenada, B.C., México.
[trujillo,olague]@cicese.mx

Evelyne Lutton
APIS Team, INRIA-Futurs,
Parc Orsay Université 4,
ORSAY Cedex, France.
evelyne.lutton@inria.fr

Francisco Fernández de
Vega
Grupo de Evolución Artificial,
Universidad de Extremadura,
Mérida, Spain.
fcfdez@unex.es

ABSTRACT

This paper describes a speciation method that allows an evolutionary process to *learn* several robot behaviors using a single execution. Species are created in behavioral space in order to promote the discovery of different strategies that can solve the same navigation problem. Candidate neurocontrollers are grouped into species based on their corresponding *behavior signature*, which represents the traversed path of the robot within the environment. *Behavior signatures* are encoded using character strings and are compared using the string edit distance. The proposed approach is better suited for an evolutionary robotics problem than speciating in objective or topological space. Experimental comparison with the NEAT method confirms the usefulness of the proposal.

Categories and Subject Descriptors

I.2.9 [Artificial Intelligence]: Robotics—*Autonomous vehicles*

General Terms

Theory, Algorithms

1. INTRODUCTION

In evolutionary robotics (ER) [3] a robot's perception-action loop is determined by a control mechanism that is artificially evolved. Behaviors emerge from the interactions of a situated agent with the environment. In ER, neural networks are used for robot control and an evolutionary algorithm (EA) is used to design and/or train the best possible NN to solve a given task. This work extends the single-solution approach by incorporating speciation into the evolutionary process. Through speciation the ER system is able to find different strategies for a single navigation problem; see [5]. When a search space is multimodal EAs employ speciation in order to find as many fitness peaks as possible. Speciation methods can be divided into two main groups. In one, EAs attempt to decompose a problem and generate specialized solutions for subproblems. In the other, EAs attempt to find problem solutions that "perform different versions of basically the same job". A noteworthy example of the latter group is the NEAT method [4], a speciating GA for neuro-evolution. NEAT allows for crossover operations between networks with different topologies. Also, NEAT simulates *incremental learning* by starting from an initial topology and incrementally adding nodes and synapses, and protects topological innovations through speciation. However, building complexity with varying topologies is of less interest if different species do

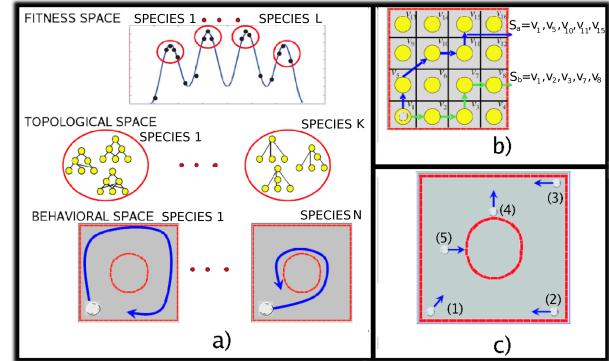


Figure 1: a) The top row shows speciation carried out in fitness space. Next, speciation based on NN topology. Finally, behavior-based speciation. b) Two sample behavior signatures. Each node is labeled, an the path consists of the string of visited nodes. c) Training environment: (1) represents the initial position for behavior signature generation; (2 - 5) each of the starting positions and headings for the training epochs. The topological map of this environment is the same as in b).

not exhibit a different functional response. The NEAT method can only guarantee a diverse set of network topologies not a diverse set of input-output mappings because two NNs can produce the same functional response even if they are topologically different. In the current work, it is hypothesized that if an appropriate comparative measure can be defined, then species will develop in different regions of behavior space, see Figure 1a.

2. BEHAVIOR-BASED SPECIATION

Speciation in behavioral space requires: 1) a behavior representation, and 2) a similarity measure. In this work, behaviors are represented through *signatures* expressed as character strings, and are compared using the string edit distance. In order to comprehend the speciation process the distinction between a behavior and an individual must be stressed. An individual is a neurocontroller x , while a behavior is the navigation strategy a induced by the sensory-motor mapping of x within an environment \mathcal{E} , written as $x \xrightarrow{\mathcal{E}} a$. A behavior is a subjective concept, while S_a represents an objective characterization of a . S_a is obtained by way of an interpretation process denoted by ψ .

Definition 1: Let x represent an individual neurocontroller and a the behavior induced by x within environment \mathcal{E} , written as $x \xrightarrow{\mathcal{E}} a$. Then, the **behavior signature** S_a represents a description of

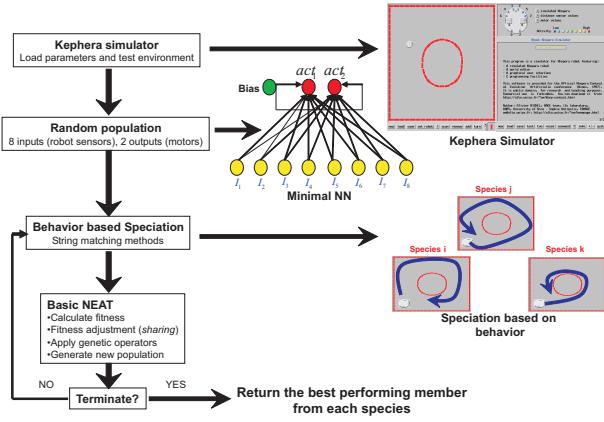


Figure 2: The ER system used to evolve several behaviors.

behavior a , obtained through a **behavior interpretation process** ψ , written as $\psi(a) \hookrightarrow S_a$.

Two examples of signatures are shown in Figure 1b. The environment is represented using a topological map where V is the set of nodes. A neurocontroller x , starting from an initial node $v_1 \in V$, will guide the robot across the map generating a path S represented by the sequence of visited nodes $S = v_i, \dots, v_n$.

N-GLD: Normalized Levenshtein distance. The Generalized Levenshtein Distance (GLD), also known as the edit distance, compares strings by various edit operations. Given an alphabet Σ and $v, u \in \Sigma$, the operations $\lambda \rightarrow v$, $v \rightarrow u$, and $u \rightarrow \lambda$, represent insertions, substitutions and deletions respectively. It is possible to define the *edit transformation* $T_{S_a, S_b} = T_1, T_2, \dots, T_l$ as a sequence of edit operations that transforms S_a into S_b . If a function $\gamma(v \rightarrow u) \geq 0$ assigns a non-negative weight to each edit operation, then the total weight of T_{S_a, S_b} is computed by $\gamma(T_{S_a, S_b}) = \sum_{i=1}^l \gamma(T_i)$. The GLD, which is a metric under certain conditions [6], is defined as $GLD(S_a, S_b) = \min \{\gamma(T_{S_a, S_b})\}$. To account for the common situation in which $|S_a| \neq |S_b|$, the normalized GLD δ_{N-GLD} for two strings $S_a, S_b \in \Sigma^*$ is given by

$$\delta_{N-GLD}(S_a, S_b) = \frac{2 \cdot GLD(S_a, S_b)}{\alpha(|S_a| + |S_b|) + GLD(S_a, S_b)}, \quad (1)$$

where $\alpha = \max \{\gamma(v \rightarrow \lambda), \gamma(\lambda \rightarrow u), v, u \in \Sigma\}$ [6].

Species behaviors: A population $\mathcal{P} = \{x_1, x_2, \dots, x_N\}$ of N neurocontrollers x , can be divided into M different species R_k ,

$$\mathcal{P} = \bigcup_{k=1}^M R_k \quad \text{where} \quad R_k \cap R_l = \emptyset \quad \text{for} \quad k \neq l. \quad (2)$$

Furthermore, let $f(x)$ represent the fitness value of neurocontroller x within environment \mathcal{E} . Then, the **species behaviors** of population \mathcal{P} within \mathcal{E} is given by the multiset $\mathcal{B} = \{a^1, \dots, a^i, \dots, a^L\}$ of L behaviors, such that $\forall a^i \in \mathcal{B}$ if $x \xrightarrow{\mathcal{E}} a^i$ and $x \in R_k$ then

$$f(x) > \sup \{f(y) \mid \forall y \in R_k, y \neq x\} \wedge f(x) > h, \quad (3)$$

where h is called the *behavior threshold*, set empirically.

The ER system. Figure 2 is a high-level view of ER system, integrated into the Kephera Simulator [1], where the parameters, EA, and training environment are loaded. The initial population contains an homogeneous collection of NN topologies. The minimal topology is a fully connected NN with 8 input neurons (sensors) and 2 output neurons (actuators). This is followed by the basic NEAT method which is a straightforward generational GA with fitness proportional selection. Species formation proceeds as follows. Given a similarity threshold γ a new individual x is added to the

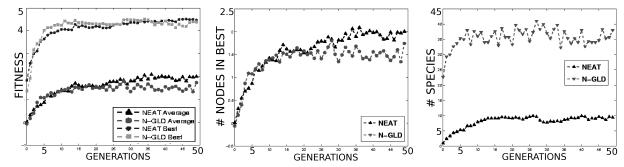


Figure 3: Performance plots (left-to-right): a) average and best fitness; b) number of nodes in best; and c) number of species.

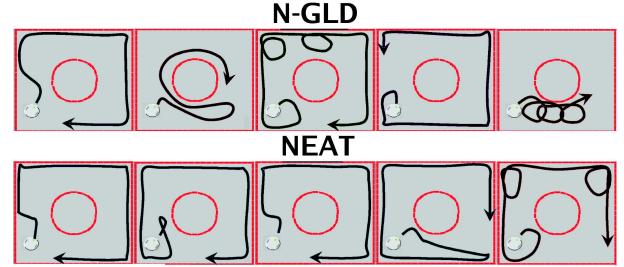


Figure 4: Species behaviors for each method.

first species B where its signature's distance to a randomly selected species member $y \in B$ is $\delta_{N-GLD}(x, y) < \gamma$. If no such species is found, then a new species A is created for x . Signatures are obtained for each NN placing the robot in node v_1 (Figure 1c). The training environment is similar to the one in [2], it offers a multimodal landscape in behavioral space. Fitness encourages behaviors where the robot navigates by: 1) moving forward in a straight line; 2) moving as fast as possible; and 3) avoiding collisions.

3. EXPERIMENTAL RESULTS

The EA parameters: # runs = 6; population size = 100; generations = 50; compatibility threshold: $\gamma_{N-GLD} = 0.4$, $\gamma_{NEAT} = 3$; behavior threshold: $h = 3.7$. Figure 3 has three performance curves, all are plotted relative to the number of generations and represent the averages over all runs. In the first graph performance is mostly equivalent. The second graph shows NEAT produces more complex individuals. In the last graph, N-GLD generates more species; populations tend to be highly diverse. Figure 4 shows five species behaviors, found in a single run for each method. In summary, the N-GLD produced a set of *species behaviors* that are unique, while NEAT could not do the same.

4. CONCLUSIONS

This paper describes a behavior-based speciation method that encourages several navigation strategies to evolve within a single population. The proposed method found several unique solutions for the same problem, while topological speciation could not.

5. REFERENCES

- [1] O. Michel. *Kephera Simulator v2 User Manual*. University of Nice-Sophia, Antipolis, 1996.
- [2] O. Migliano, H. H. Lund, and S. Nolfi. Evolving mobile robots in simulated and real environments. *Artificial Life*, 2(4):417–434, 1995.
- [3] S. Nolfi and D. Floreano. *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. Bradford Book, 2004.
- [4] K. O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127, 2002.
- [5] L. Trujillo, G. Olague, E. Lutton, and F. Fernández. Discovering several robot behaviors through speciation. In *Proceedings of EvoWorkshops 2008*, Lecture Notes in Computer Science, pages 164–174. Springer, 2008.
- [6] L. Yujian and L. Bo. A normalized levenshtein distance metric. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(6):1091–1095, 2007.