

# Discovering Several Robot Behaviors through Speciation

Leonardo Trujillo<sup>1</sup>, Gustavo Olague<sup>1</sup>,  
Evelyne Lutton<sup>2</sup>, and Francisco Fernández de Vega<sup>3</sup>

<sup>1</sup> EvoVisión Project, CICESE Research Center, Ensenada, B.C., México  
trujillo@cicese.mx, olague@cicese.mx

<sup>2</sup> APIS Team, INRIA-Futurs, Parc Orsay Université 4, ORSAY Cedex, France  
evelyne.lutton@inria.fr

<sup>3</sup> Grupo de Evolución Artificial, Universidad de Extremadura, Mérida, Spain  
fcofdez@unex.es

**Abstract.** This contribution studies speciation from the standpoint of evolutionary robotics (ER). A common approach to ER is to design a robot's control system using neuro-evolution during training. An extension to this methodology is presented here, where speciation is incorporated to the evolution process in order to obtain a varied set of solutions for a robotics problem using a single algorithmic run. Although speciation is common in evolutionary computation, it has been less explored in behavior-based robotics. When employed, speciation usually relies on a distance measure that allows different individuals to be compared. The distance measure is normally computed in objective or phenotypic space. However, the speciation process presented here is intended to produce several distinct robot behaviors; hence, speciation is sought in behavioral space. Thence, individual neurocontrollers are described using *behavior signatures*, which represent the traversed path of the robot within the training environment and are encoded using a character string. With this representation, *behavior signatures* are compared using the normalized Levenshtein distance metric (N-GLD). Results indicate that speciation in behavioral space does indeed allow the ER system to obtain several navigation strategies for a common experimental setup. This is illustrated by comparing the best individual from each species with those obtained using the Neuro-Evolution of Augmenting Topologies (NEAT) method which speciates neural networks in topological space.

## 1 Introduction

Evolutionary Robotics (ER) [1] can be seen as an extension to behavior-based robotics (BBR) [2,3]. In classic BBR behaviors are hand-designed by a human expert. On the other hand, in ER the sensory-motor mappings that control the way in which a robot interacts with its surroundings emerge from an artificial evolutionary process. Consequently, ER encourages robot behaviors to emerge from complex interactions between: 1) the autonomous agent; 2) the control mechanism; and 3) the physical environment. ER employs evolutionary computation (EC) methods in the design process of artificial neural networks (ANN) that provide the control mechanism for an autonomous robot. When using ER techniques, most researchers are only interested in finding a single solution for the problem at hand, e.g. a navigation strategy. However, using evolution

to find a single *super* individual can have several disadvantages [4]. For instance, a large amount of computational effort is not exploited because only one solution from the population is used. Moreover, populations can converge prematurely and solutions may become overfitted to the training problem instance. A workaround to the previous shortcomings is to employ diversity preservation methods. For instance, speciation allows individuals to compete within their own species instead of the entire population. In this way, novel but perhaps less apt solutions can still propagate their genetic material and populations can stay in a more heterogeneous state. Therefore, through speciation a diverse set of solutions could conceivably be obtained from a single evolutionary run, even when all the individuals are trained using the same environment in an ER system.

**Outline of the Proposed Approach.** This work introduces a behavior-based speciation method, where the behavior exhibited by each neurocontroller is described by what are called *behavior signatures* which allow different behaviors to be compared. Behavior signatures are given as character strings that contain the path followed by the robot within a topological, or graph-based, representation of the environment. Hence, a string similarity measure can be used to compare behavior signatures, in this case the normalized Levenshtein distance metric (N-GLD) is proposed [5]. Thence, speciation can be carried out in behavioral space, a more natural approach for BBR than using objective or phenotypic space to speciate, both of which are more prevalent in other EC problem domains; see Figure 1a. The technique promotes the emergence of distinct robot behaviors, each following a different navigation strategy within the same training environment. The speciation strategy is incorporated within the Neuro-Evolution of Augmenting Topologies (NEAT) method [6]. NEAT adds ANN complexity in an incremental manner and evolves the topology and connection weights concurrently.

This paper proceeds as follows: Section 2 describes the basic concept of speciation and reviews related work. Section 3 introduces the proposed speciation method. Implementation details and the evolutionary setup are discussed in Section 4. Experimental results are presented in Section 5. Finally, in Section 6 concluding remarks are given.

## 2 Speciation and Diversity Preservation

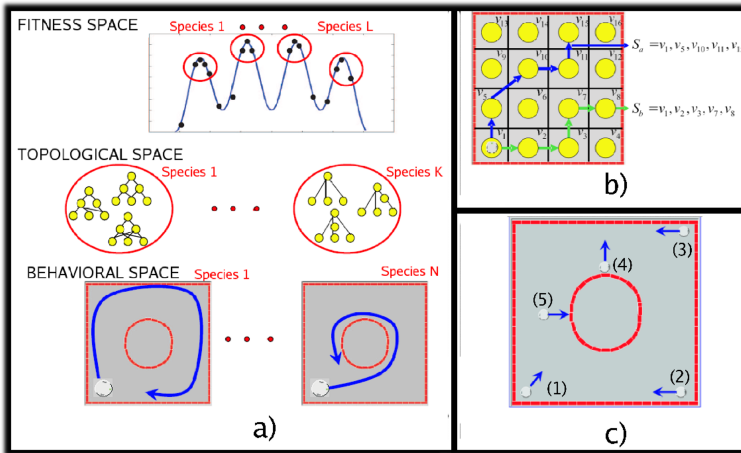
When a multimodal space exists, it may be desirable to find as many solutions as possible. To achieve this goal, a common approach within EC is to incorporate a speciation mechanism within the evolutionary algorithm (EA) of choice.

In formal terms, a speciating algorithm applies a mechanism  $\mathcal{D}$  that maximizes the diversity of individuals within a population  $\mathcal{P}$ , and also maintains a high mean population fitness  $\mathcal{F}(\mathcal{P})$ . Therefore, an idealized mechanism would imply that,

$$\mathcal{D}(\mathcal{P}) \longrightarrow \max \{H(\mathcal{P})\} \wedge \max \{\mathcal{F}(\mathcal{P})\} , \quad (1)$$

where  $H(\mathcal{P})$  is the entropy of population  $\mathcal{P}$ . Some speciation methods are of general use like fitness sharing [7], while others are domain specific such as symbiosis [8].

**Related Work.** Speciating methods can be grouped into two classes. The first contains techniques that perform problem decomposition and specialization, what some



**Fig. 1.** a) The top row shows the basic niching technique carried out in fitness space. Next, speciation based on topological similarities between ANNs (NEAT). Finally, the proposed behavior-based speciation. b) Two sample behavior signatures generated in the topological map representation. Each node is labeled, and the path consists of the string of visited nodes by the robot. c) Training environment used: (1) represents the initial position for behavior signature generation; (2 - 5) each of the starting positions and headings for the four training epochs. The topological representation of this environment is the same as in b) using a  $4 \times 4$  grid.

researchers call “evolutionary divide and conquer” [9]. Some examples include the work by Moriarty & Mikkulainen [8] and Dunn *et al.* [9]. The second group performs speciation in order to find problem solutions that “perform different versions of basically the same job” [10]. In other words, each individual represents a complete stand-alone solution, and each species contains solutions with distinctive properties. Relevant examples include the work by Hocoğlu & Sanderson [11] and Stanley & Miikkulainen [6]. Hocoğlu & Sanderson evolve alternative paths in 2D and 3D environments. However, their problem formulation is given in terms of a deliberative control mechanism, as apposed to the BBR approach of ER. Stanley & Miikkulainen introduce the NEAT method, a specialized GA that uses speciation to obtain alternative ANNs. NEAT evolves both the topology and connection weights, thus carrying out incremental learning of network complexity. NEAT has shown an ability to solve *hard* problems, hence, it is used as the basis for the proposed ER system.

**Neuro-Evolution of Augmenting Topologies.** The NEAT method introduces several advantages when compared with other neuro-evolution systems. For instance, the encoding used allows for crossover operations to be carried out between networks with different topologies. NEAT simulates *incremental learning* by starting from an initial topology and incrementally adding new nodes and synapses. Finally, NEAT protects topological innovations with speciation. To the authors knowledge, NEAT has not been used in BBR problems, marking the present work as the first such instance.

**Speciation in NEAT.** Speciation in NEAT groups ANN using a measure of topological similarity. The method defines a similarity measure between two ANN chromosomes using the number of disjoint genes, excess genes, and connection weight differences; where genes can represent a network node or a synaptic weight. Therefore, a measure of similarity  $\delta_{NEAT}$  is given by

$$\delta_{NEAT} = \frac{c_1 \cdot G + c_2 \cdot D}{N} + c_3 \cdot \overline{W}, \quad (2)$$

where  $G$  is the number of excess genes,  $D$  the number of disjoint genes,  $\overline{W}$  the average weight difference of matching genes,  $c_x$  are weight coefficients normally set to  $c_1 = c_2 = 1$  and  $c_3 = 0.4$ , and  $N$  a normalization factor; for details see [6]. Thus, given a similarity threshold  $\delta_t$  a new individual  $a$  is added to the first species  $B$  where its distance  $\delta_{NEAT}$  to a randomly selected species member  $b \in B$  is  $\delta_{NEAT}(a, b) < \delta_t$ . If no such species is found, then a new species  $A$  is created for  $a$ . Explicit fitness sharing is used within each species. The adjusted fitness  $f'_i$  for the individual  $i$  is calculated according to its distance  $\delta$  to every other individual  $j$  in the population,

$$f'_i = \frac{f_i}{\sum_{j=1}^n sh(\delta(i, j))}, \quad (3)$$

where function  $sh(\delta(i, j))$  is set to 0 if  $\delta(i, j) \geq \delta_t$  and 1 otherwise.

**Limitations of Topological Speciation.** The goal behind speciation is to produce a functionally diverse set of solutions. Building complexity with varying topologies is less interesting if different species do not exhibit an appreciable difference in their functional response. The speciation mechanism proposed by NEAT can only guarantee a diverse set of network topologies not a diverse set of functional solutions. This can be understood with the concept of competing conventions [12], because two ANNs can produce the same functional response even when they are topologically different. In the present 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 1.

### 3 Behavior-Based Speciation

In order to be able to speciate in behavior space an appropriate behavior representation is necessary along with a proper comparative measure. This work presents a behavior representation that employs *signatures* expressed as character strings. Thus, similarity measures are taken from string comparison techniques.

**Behaviors and Neurocontrollers.** The distinction between a behavior and an individual in the evolutionary process must be stressed because they do not represent the same concept. An individual represents a particular neurocontroller  $x$ , while a behavior is a navigation strategy  $a$  induced by the sensory-motor mapping of  $x$  within an environment  $\mathcal{E}$ , written as  $x \xrightarrow{\mathcal{E}} a$ . Moreover, due to competing conventions a many-to-one relationship should be assumed between individuals and behaviors. Consequently, let

two individuals  $x$  and  $y$  induce behaviors  $x \xrightarrow{\mathcal{E}} a$  and  $y \xrightarrow{\mathcal{E}} a$  respectively. The notation implies that the underlying navigation strategy  $a$  is shared by both  $x$  and  $y$ . Also, it is assumed that each individual neurocontroller  $x$  induces one and only one behavior within  $\mathcal{E}$ . Furthermore, a behavior is considered to be a subjective concept, while its corresponding signature  $S_a$  represents an objective characterization of  $a$ . It can be said that  $S_a$  is obtained by way of an interpretation process denoted by  $\psi$ .

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

Indeed, making measurements of specific attributes of a behavior is common, however the same cannot be trivially done for the behavior itself. The reason for this is that  $\psi$  is an attempt to interpret a behavior as if it had concrete existence, when in fact it represents an abstract concept. In this work,  $\psi$  is such that  $S_a$  represents the traversed path of the robot within  $\mathcal{E}$ . Note that the proposed speciation method works under the assumption that each behavior  $a$  is characterized by one and only one signature  $S_a$ .

Figure 1b gives a graphical representation of the proposed behavior signatures. The environment is represented using a topological map  $\mathcal{M} = (V, E)$  where  $V$  is the set of nodes in  $\mathcal{M}$  and  $E$  the set of edges. 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 nodes visited by the robot  $S = v_i, \dots, v_j, \dots, v_n$ . In order to obtain a signature  $S$ , a controller  $x$  navigates the robot for 4000 cycles, and the position of the robot is updated every 10 cycles. If at a given update cycle  $t$ , the node  $v^t$  that the robot occupies is different from the node it occupied at the previous update cycle  $v^{t-1}$ , then  $v^t$  is added to  $S$ . To avoid having the same initial nodes in all behavior signatures, which could influence the similarity measure, nodes are added to  $S$  only after an initial stabilizing time period of 500 cycles. The stabilizing time eliminates nodes from  $S$  that all behavior signatures would have as their leading characters due to the shared starting position and not due to any meaningful similarity. Because  $S$  is a character string a string similarity measure  $\delta(S_a, S_b)$  can be applied to compare different signatures. Therefore,  $\delta(S_a, S_b)$  defines a distance between behaviors  $a$  and  $b$ .

**N-GLD: Normalized Levenshtein Distance.** Before describing the N-GLD metric some preliminary definitions must be established. The alphabet is  $\Sigma$ ,  $\Sigma^*$  is the set of strings over  $\Sigma$ , and  $\lambda \notin \Sigma$  is the null string. Here,  $\Sigma = V$  and  $\Sigma^*$  is the set of possible paths in  $\mathcal{M}$ . A string  $S \in \Sigma^*$  is expressed as  $S = s_1, s_2, \dots, s_n$ , where  $s_i \in \Sigma$  is the  $i$ th symbol of  $S$ , and  $|S| = n$  the size of the string (the null string has  $|\lambda| = 0$ ). The Generalized Levenshtein Distance (GLD), also known as the edit distance, compares strings by various edit operations, commonly using the deletion, insertion, and substitution of individual symbols [5]. If  $v, u \in \Sigma$ , an elementary edit operation is defined as a pair  $(v, u) \neq (\lambda, \lambda)$ , and is written as  $v \rightarrow u$ , where  $|v|, |u| \in \{0, 1\}$ . 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 weight 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  $\gamma(T_{S_a, S_b}) = \sum_{i=1}^l \gamma(T_i)$ , and the GLD is defined as,

$$GLD(S_a, S_b) = \min \{ \gamma(T_{S_a, S_b}) \} . \quad (4)$$

The GLD is a metric over  $\Sigma^*$  if :

1.  $\forall v, u \in \Sigma \cup \{ \lambda \} , \gamma(v \rightarrow v) = 0 .$
2.  $\gamma(v \rightarrow u) > 0$  if  $(v \neq u) \wedge [ \gamma(v \rightarrow u) = \gamma(u \rightarrow v) ] .$

In order to account for the common situation in which  $|S_a| \neq |S_b|$ , a normalized version of GLD is required. Yuijian and Bo [5] define the normalized GLD  $\delta_{N-GLD}$  for two strings  $S_a, S_b \in \Sigma^*$  as

$$\delta_{N-GLD}(S_a, S_b) = \frac{2 \cdot GLD(S_a, S_b)}{\alpha(|S_a| + |S_b|) + GLD(S_a, S_b)} , \quad (5)$$

where  $\alpha = \max \{ \gamma(v \rightarrow \lambda), \gamma(\lambda \rightarrow u), v, u \in \Sigma \}$ , and  $\delta_{N-GLD}(\lambda, \lambda) = 0$ .

It was shown in [5] that the  $\delta_{N-GLD}$  has the following properties:

1. It satisfies  $0 \leq \delta_{N-GLD}(S_a, S_b) \leq 1$ .
2.  $\delta_{N-GLD}(S_a, S_b) = 0$  if and only if  $S_a = S_b$ .
3. It is symmetric, because  $\delta_{N-GLD}(S_a, S_b) = \delta_{N-GLD}(S_b, S_a)$ .
4. It satisfies the triangle inequality, thence, it is a metric over  $\Sigma^*$  if,  $\forall v \in \Sigma, \gamma(v \rightarrow \lambda) = \gamma(\lambda \rightarrow v) = \alpha$ , and  $\gamma$  is a metric over the set of elementary operations.

In [5] the following weight function is suggested, and is used in the present work:

$$\gamma(v, v) = 0, \gamma(v, u) = 1, \text{ and } \gamma(v, \lambda) = \gamma(\lambda, u) = 1 \forall v, u \in \Sigma.$$

**Species Behaviors.** Before presenting the experimental setup, another domain specific concept is defined that will facilitate further discussion of the proposed method.

**Definition 2.** A population  $\mathcal{P} = \{x_1, x_2 \dots x_j \dots x_N\}$  of  $N$  neurocontrollers  $x$ , can be divided into  $M$  different species  $R_k$  with  $k = 1 \dots M$ , such that

$$\mathcal{P} = \bigcup_{k=1}^M R_k \text{ where } R_k \cap R_l = \emptyset \text{ for } k \neq l . \quad (6)$$

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

$$f(x) > \sup \{ f(y) \mid \forall y \in R_k, y \neq x \} \wedge f(x) > h , \quad (7)$$

where  $h$  is called the behavior threshold which is set empirically.

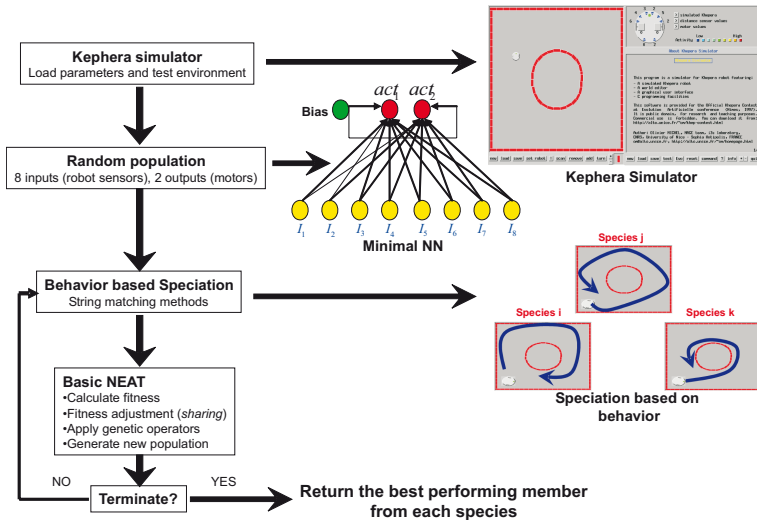
Therefore, every  $a^i \in \mathcal{B}$  is induced by one and only one neurocontroller  $x \in \mathcal{P}$ , and every such neurocontroller is the super-individual of its corresponding species. Given

Definition 2, it is possible to observe that *species behaviors* are contingent on the environment  $\mathcal{E}$  that the neurocontrollers interact with. In the general ER framework,  $\mathcal{E}$  refers to the training environment employed. An ER system that produces a large  $\mathcal{B}$  is said to have found several super-individuals. However, it cannot be assumed that these behaviors represent distinctively different navigation strategies. Therefore, an objective evaluation must be performed in order to determine which of the members of  $\mathcal{B}$  do indeed represent “different versions of basically the same job”.

### 4 Implementation of the ER System

This section first describes the Kephera robot, outlines the ER algorithm and gives details on the training environment and fitness function employed.

**The Kephera Robot and Simulator.** The Kephera is very common within the ER community, it possesses a simple structure and control mechanism that makes it ideal to test novel methods. The Kephera has two DC motors  $act_1$  and  $act_2$  as actuators, and eight infrared proximity sensors  $I_1, I_2, \dots, I_8$ . Evolving neurocontrollers on-line on a real Kephera robot can be quite cumbersome and problematic [1]. Therefore, much of the ER research is conducted on a simulated environment. Robot and environment simulation in the present work is done on the freeware Kephera Simulator version 2.0



**Fig. 2.** An overview of the ER system used to evolve alternatives behaviors. First, the Kephera Simulator loads all the algorithm parameters and acts as the interface with the user. Next, the minimal topology of the ANNs used for control. Followed by the neuro-evolutionary system, beginning with the behavior-based speciation process that groups ANNs according to their behavior signatures at each generation. The last steps are basic GA processes, with special genetic operators used by the NEAT method. Finally, a representative Neurocontroller from each species is obtained, the set of *Species Behaviors*  $\mathcal{B}$ .

[13]. The simulator gives a satisfactory modeling of the physical properties of a real Kephra robot, and the ability to write any kind of control algorithms in C or C++.

**The ER System for Behavior-Based Speciation.** Figure 2 is a high-level view of the algorithm, based on the NEAT<sup>1</sup> [6] method and employing the Kephra simulator [13]. The ER system is integrated into the Kephra Simulator where the robot parameters, EA, and training environment are loaded. The initial population contains an homogeneous collection of ANN topologies. The minimal topology is a fully connected ANN with 8 input neurons (one for each sensor) and 2 output neurons (for  $act_1$  and  $act_2$ ) with randomly assigned weights. This is followed by the basic NEAT method which is a straightforward generational GA with fitness proportional selection. The only additional mechanism is that related to speciation and fitness adjustment. The basic process of speciation in NEAT was described in Section 2. The main difference with the proposed behavior-based speciation is the use of behavior signatures with a similarity measure based on the N-GLD metric. Signatures are obtained for each ANN placing the robot in node  $v_1$  at a  $45^\circ$  heading, see Figure 1c.

**Training Environment.** The training environment is very similar to the one used in [1] shown in Figure 1c. It is simple, basically a square room with a “big” obstacle in the middle. In spite of this, the environment offers a multimodal landscape in behavioral space where different navigation strategies are possible.

**Fitness Evaluation.** The type of behavior that simulated evolution *should* be searching for is one where the robot navigates around the environment exhibiting the following properties: 1) the robot moves forward in a straight line; 2) the robot moves as fast as possible; and 3) the robot avoids collisions. For these properties to emerge, fitness is assigned as in [14], where for an individual neurocontroller  $x$ ,

$$f(x) = \frac{1}{N \cdot M} \sum_{j=1}^M \sum_{k=1}^N V_i (1 - \sqrt{\Delta v_k}) (1 - \varphi_k), \quad (8)$$

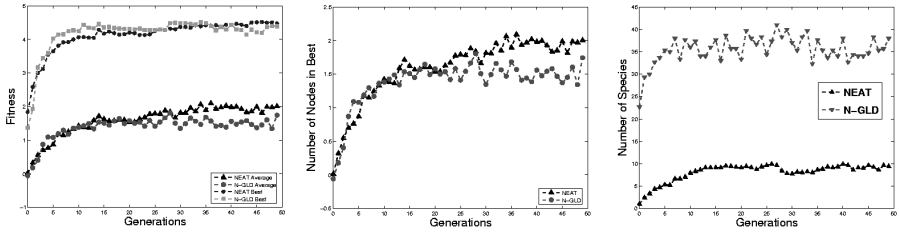
where  $V_k$  is the sum of the two motor speeds at time step  $k$ .  $\Delta v_k$  is the absolute difference between the two motors.  $\varphi$  is the normalized activation value of the infrared sensor with the highest activation. Moreover,  $M$  is the number of test runs, or epochs, and  $N$  the total number of time steps or cycles within an environment during an epoch  $j$ . The number of epochs is set to  $M = 4$  with the initial position and heading of the robot for each epoch shown in Figure 1c, while the number of cycles per epoch is  $N = 3000$ . The fitness function  $f(x)$  is maximized with better performance.

## 5 Experimental Results

This section describes the results of the proposed speciation method and how it compares with the NEAT method. The parameters employed by each method are the following: number of runs = 6; population size = 100; generations = 50; crossover rate = 0.75;

<sup>1</sup> Source code downloaded from the web site of the Neural Networks Research Group of the University of Texas at Austin: <http://www.cs.utexas.edu/nn/>



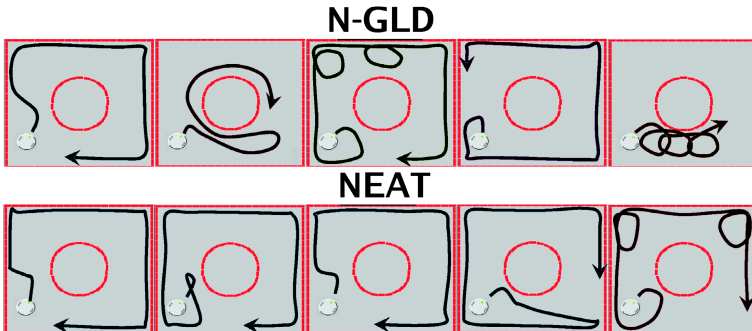


**Fig. 3.** Performance plots that compare NEAT and the proposed behavior speciation method with the N-GLD metric. The plots, from left-to-right, are: a) average population and best individual fitness; b) number of nodes in best solution; and c) number of species.

compatibility threshold (used to assign species membership):  $\delta_{N-GLD} = 0.4$  ,  $\delta_{NEAT} = 3$ ; behavior threshold (used to identify *species behaviors*):  $h = 3.7$ . Both methods share all the run-time parameters except for the compatibility threshold  $\delta_t$ , which was set experimentally for N-GLD and as in [6] for NEAT.

Figure 3 presents three comparative performance curves. All graphs are plotted relative to the number of generations and represent averages over the total number of runs. The plots, from left-to-right, are: a) average population and best individual fitness; b) number of nodes in best solution; and c) number of species. In the first graph, performance is mostly equivalent between both methods. NEAT performs slightly better in average performance which suggests that solutions are better fitted, or overfitted, to the training environment. In the second, the number of nodes of the best individual shows that the NEAT method produces more complex individuals. However, these larger individuals do not yield a higher fitness. With regards to the number of species, the NEAT method produces a lower number of species than does the N-GLD measure. Therefore, the proposed speciation method keeps a more diverse set of solutions.

Additionally, Figure 4 presents two sets of species behaviors  $\mathcal{B}$ , one each for the NEAT topological measure and the N-GLD measure. Each of the six runs produced a corresponding  $\mathcal{B}$ , however only one of them is shown due to the length constraints of



**Fig. 4.** The set of species behaviors  $\mathcal{B}$  found for each of the compared speciation methods: Top row, behavior-based speciation with N-GLD metric; bottom row NEAT’s topological speciation

the paper. Nevertheless, the  $\mathcal{B}$  shown for each are highly representative, and any further discussion derived from these behaviors generalizes well to the other sets of results. The behavior-based speciation with G-NLD produced *species behaviors* that are all unique. Each have a different manner in which they perform navigation within the environment. Therefore, every behavior represents a qualitatively different solution from the rest. On the other hand, NEAT's topology-based speciation fails to obtain the same degree of diversity. In this case, only one of the species behaviors is different from the rest. Therefore, most NEAT species converge to very similar navigation strategies. In sum, behavior-based speciation was indeed able to find solutions that "perform different versions of basically the same job", while topological speciation fails in this task.

## 6 Conclusions

In an ER system, obtaining several behaviors could provide a better characterization of the space of possible solutions because the same tasks can usually be performed using different behaviors. Thence, a system that is capable of obtaining several solutions from a single evolving population is of interest. The present work describes a novel behavior-based speciation method that encourages several navigation strategies to evolve concurrently within a single evolutionary process. Behaviors are compared using their *signatures*, which represent a traversed path across the training environment. A similarity measure employing the string edit distance is proposed, the N-GLD metric. This measure is incorporated into the NEAT method, substituting, and subsequently compared with, NEAT's topology-based similarity measure. Results indicate that the EA was able to produce several different navigation strategies using the proposed behavior-based speciation; the same could not be achieved using NEAT's similarity measure. This work presents the first instance within ER literature where various navigation strategies are evolved concurrently, thus providing several strategies from which the end user can choose from. Finally, future work should focus on how to relax the two main assumptions made within the proposed speciation method, namely: 1) that each neurocontroller induces one and only one behavior; and, 2) that each behavior can be instantiated by one and only one signature.

**Acknowledgements.** Research funded by the Ministerio de Educación y Ciencia (project Oplink - TIN2005-08818-C04), the LAFMI project, and the Junta de Extremadura Spain. First author supported by scholarship 174785 from CONACyT México.

## References

1. Nolfi, S., Floreano, D.: Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines. Bradford Book (2004)
2. Brooks, R.A.: A robust layered control system for a mobile robot. IEEE Journal of Robotics and Automation 2(1), 14–23 (1986)
3. Brooks, R.A.: Intelligence without representation. Artif. Intell. 47(1-3), 139–159 (1991)
4. Yao, X.: Evolving artificial neural networks. PIEEE: Proceedings of the IEEE 87(9), 1423–1447 (1999)

5. Yujian, L., Bo, L.: A normalized levenshtein distance metric. *IEEE Trans. Pattern Analysis and Machine Intelligence* 29(6), 1091–1095 (2007)
6. Stanley, K.O., Miikkulainen, R.: Evolving neural networks through augmenting topologies. *Evolutionary Computation* 10(2), 99–127 (2002)
7. Goldberg, D.E., Richardson, J.: Genetic algorithms with sharing for multimodal function optimization. In: *Proceedings of the Second International Conference on Genetic Algorithms and their application*, pp. 41–49. Lawrence Erlbaum Associates, Inc, Mahwah (1987)
8. Moriarty, D.E., Mikkulainen, R.: Efficient reinforcement learning through symbiotic evolution. *Machine Learning* 22(1-3), 11–32 (1996)
9. Dunn, E., Olague, G., Lutton, E.: Parisian camera placement for vision metrology. *Pattern Recogn. Lett.* 27(11), 1209–1219 (2006)
10. Darwen, P.J., Yao, X.: Speciation as automatic categorical modularization. *IEEE Trans. Evolutionary Computation* 1(2), 101–108 (1997)
11. Hocaoglu, C., Sanderson, A.C.: Planning multiple paths with evolutionary speciation. *IEEE Trans. Evolutionary Computation* 5(3), 169–191 (2001)
12. Montana, D.J., Davis, L.: Training feedforward neural networks using genetic algorithms. In: Sridharan, S. (ed.) *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pp. 762–767. Morgan Kaufmann, San Francisco (1989)
13. Michel, O.: *Khepera Simulator v2 User Manual*, University of Nice-Sophia, Antipolis (1996)
14. Miglino, O., Lund, H.H., Nolfi, S.: Evolving mobile robots in simulated and real environments. *Artificial Life* 2(4), 417–434 (1995)