J Intell Robot Syst (2011) 64:323–351
DOI 10.1007/s10846-011-9542-z

Speciation in Behavioral Space for Evolutionary Robotics

Leonardo Trujillo · Gustavo Olague · Evelyne Lutton · Francisco Fernández de Vega · León Dozal · Eddie Clemente

Received: 9 September 2010 / Accepted: 9 January 2011 / Published online: 20 January 2011

Abstract In Evolutionary Robotics, population-based evolutionary computation is used to design robot neurocontrollers that produce behaviors which allow the robot to fulfill a user-defined task. However, the standard approach is to use canonical evolutionary algorithms, where the search tends to make the evolving population converge towards a single behavioral solution, even if the high-level task could be accomplished by structurally different behaviors. In this work, we present an

L. Trujillo (✉)
Instituto Tecnológico de Tijuana, Av. Tecnolǵico S/N, Fracc. Tomás Aquino,
Tijuana, BC, México
e-mail: leonardo.trujillo.ttl@gmail.com

G. Olague · L. Dozal
Departamento de Ciencias de la Computación, División de Física Aplicada,
Centro de Investigación Científica y de Educación Superior de Ensenada,
Proyecto Evovisión, Km. 107 Carretera Tijuana-Ensenada, 22860, Ensenada, BC, México

G. Olague
e-mail: olague@cicese.mx

L. Dozal
e-mail: lfdozal@hotmail.com

E. Lutton
AVIZ Team at INRIA Saclay Ile de-France, Bat. 490, Université Paris-Sud,
91405 Orsay CEDEX, France
e-mail: evelyne.lutton@inria.fr

F. Fernández de Vega
Grupo de Evolución Artificial, Universidad de Extremadura, Centro Universitario de Mérida
C/Sta Teresa de Jornet, 38, 06800, Merida, Spain
e-mail: fcofdez@unex.es

E. Clemente
Tecnológico de Estudios Superiores de Ecatepec, Ave. Carlos Hank Gonzalez Esq. Ave.
Tecnológico S/N, Col. Valle de Anahuac, Ecatepec de Morelos, Edo. de México, México
e-mail: eclemente@tese.edu.mx

approach that preserves behavioral diversity within the population in order to produce a diverse set of structurally different behaviors that the robot can use. In order to achieve this, we employ the concept of speciation, where the population is dynamically subdivided into sub-groups, or species, each one characterized by a particular behavioral structure that all individuals within that species share. Speciation is achieved by describing each neurocontroller using a representations that we call a behavior signature, these are descriptors that characterize the traversed path of the robot within the environment. Behavior signatures are coded using character strings, this allows us to compare them using a string similarity measure, and three measures are tested. The proposed behavior-based speciation is compared with canonical evolution and a method that speciates based on network topology. Experimental tests were carried out using two robot tasks (navigation and homing behavior), several training environments, and two different robots (Khepera and Pioneer), both real and simulated. Results indicate that behavior-based speciation increases the diversity of the behaviors based on their structure, without sacrificing performance. Moreover, the evolved controllers exhibit good robustness when the robot is placed within environments that were not used during training. In conclusion, the speciation method presented in this work allows an evolutionary algorithm to produce several robot behaviors that are structurally different but all are able to solve the same robot task.

Keywords Evolutionary robotics · Speciation · Behavioral space

Mathematics Subject Classification (2010) 68T40

1 Introduction

Evolutionary robotics (ER) is the application of evolutionary computation during the design of various components of a robotic system, such as the control system or the morphology of the robot. This work deals with the problem of evolving artificial neural networks (ANNs) that perform the reactive control for an autonomous robot (neurocontrollers) [22], a process known as neuroevolution. In this context, the main characteristic of ER is that the evolutionary search is guided by the behavioral performance of the robot when it is left free to act within an environment [7]. The search does not favor neurocontrollers that produce a specific behavioral structure, it searches for behaviors that fulfill general criteria related to the task that is being addressed; for example, for a navigation problem the search might be biased towards behaviors that help the robot avoid obstacles or explore the environment [22]. In other words, the evaluation of individual neurocontrollers does not consider how a given task ought to be performed, it only rates the behavioral outcome of the control process. In fact, for some robot tasks it is reasonable to assume that structurally different behaviors could achieve the same goal in different ways; i.e., many problems are multi-modal. For instance, if a robot needs to navigate the length of a corridor, it could use either wall as reference or attempt to stay in the middle. In this example, all three behaviors are different but all achieve the same high-level goal. However, as stated above, canonical ER research does not address the multi-modality of many robotic tasks that is evident from a behavioral standpoint.

In evolutionary computation (EC), multi-modal problems are solved by incorporating mechanisms that promote population diversity, such as speciation [9, 13]. Speciation in the context of EC was inspired by the Neo–Darwinian phenomenon that occurs in biology, where different varieties emerge from a single population of evolving individuals. Each variety is essentially a sub-group within the population, where all of the individuals share a common trait that individuals from other sub-groups lack. Over time, the differences will make the sub-groups diverge to the point that they are be regarded as distinct species, all descendant from the same initial population. This phenomenon has been exploited in EC because speciation allows an evolutionary algorithm to maintain diversity throughout the entire search, thus generating a variety of individual solutions and identifying several local optima. While this process happens naturally in biology, in EC it is induced by special modifications of the basic evolutionary process. For instance, a measure of similarity must be defined in order to determine species membership; i.e., there must be some way of establishing whether or not two individuals share the same salient traits.

1.1 Outline of the Proposal and Contributions

This work is concerned with the development of a speciation method for ER that allows an evolutionary algorithm to find a structurally diverse set of robot behaviors that perform the same task, exploiting the multi-modality of the problem. To achieve this goal, we propose a measure of *similarity* based on behavioral characteristics, see Fig. 1. For example, current speciating algorithms focus on genotypic similarities,

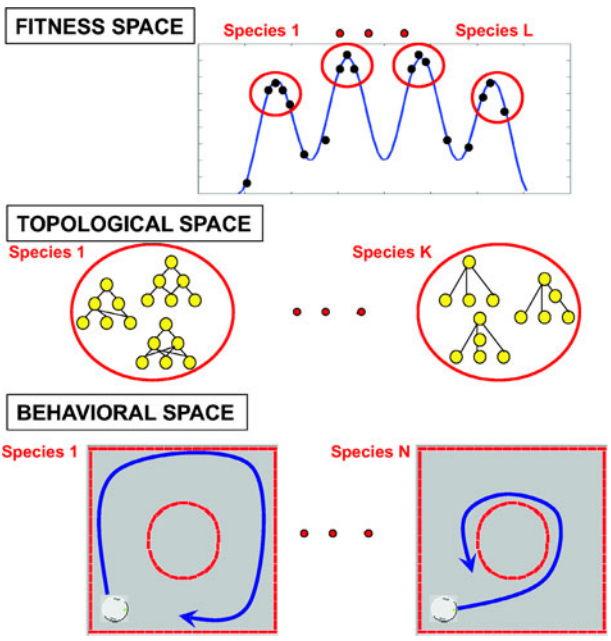


Fig. 1 Speciation in neuroevolution can take place in different spaces for ER. The *top row* shows the basic niching technique carried out in fitness space, common in MOEAs. Next, speciation based on topological similarities between ANNs. Finally, the *bottom row* shows the proposed behavior-based speciation where each species has distinct behavioral traits

while others consider the objective space of a problem [1]. However, none of these spaces can be used to directly measure the type of diversity that is of interest in ER, namely behavioral diversity. For instance, if one wishes to measure the behavioral outcome for the navigation problem described above, then the robot that stays close to a wall and the robot that stays in the middle of the corridor could receive the same objective score. Therefore, even if their behaviors are different, this dissimilarity would not be expressed in the space of the objective function. In the case of phenotypic or genotypic space, there are instances in which a high amount of diversity, or lack thereof, does not guarantee a similar diversity in behavioral space. This claim is particularly true for the evolution of neurocontrollers, where topologically different ANNs can produce the same functional response [18].

In this work a novel perspective is taken, robot controllers are analyzed in the space of possible behaviors. Each neurocontroller within the evolving population is represented using a *behavior signature*, which describes the behavior that each neurocontroller produces. Behavior signatures represent the path that the robot follows when it is placed within the environment, and are coded using character strings. Using this representation, it is possible to compare neurocontrollers based on the signatures they generate by computing a measure of string similarity; three such measures are tested, the Normalized Generalized Levenshtein Distance Metric [30, 32], a measure of Linguistic Complexity [15]; and a string similarity measure based on the largest common subsequence. In this way, we implement a speciation method that generates species which exhibit unique behavioral traits. Our proposal is validated by experimental tests on two robotic tasks, using several training environments and two different robots. Moreover, comparisons are carried out between a canonical genetic algorithm, the topology-based speciation of the NEAT method [29], and the proposed behavior-based speciation. The robustness of the evolved behaviors was tested on a more complex and unknown environment, and a descriptive analysis of the evolved behaviors is presented to illustrate the behavioral diversity achieved by our approach [14]. Finally, the scalability of our method has been confirmed by transferring the evolved behaviors onto a real robot, a Pioneer P2-AT.

The paper proceeds as follows. Section 2 reviews the topic of speciation and Section 3 introduces our behavior-based speciation. The experimental setup is presented in Section 4 and a discussion of our results is provided in Section 5. Finally, Section 6 contains our concluding remarks.

2 Evolutionary Computation and Speciation

Evolutionary computation encompasses a wide variety of population-bases metaheuristics for search and optimization based on the principles of Neo–Darwinian evolution [3]. In an evolutionary algorithm, an individual represents a possible solution to a problem, and an objective function is used to characterize the performance of each individual (fitness). Through a stochastic mechanism of selection based on performance, new individuals are created (offspring) by stochastically combining (crossover) and modifying (mutation) those selected individuals (parents). This process is repeated a fixed number of iterations (generations), and the best individual

found is returned as the solution. The most widely known evolutionary technique is the genetic algorithm, and it is the method used in this work. However, there is a wide variety of evolutionary approaches in current literature, which include multiobjective techniques [1], vector-based methods [26] and memetic approaches [20, 23], as well as other population based heuristics that are inspired in biological processes [4, 6].

When artificial evolution is used to search within a multi-modal space, it is often advantageous to explicitly promote diversity in order to find each of the fitness peaks. A common way to achieve this, is to incorporate a speciation mechanism that attempts to maximize the diversity of individuals, while maintaining a high population fitness [9, 13]. Some methods are of general use like fitness sharing [9, 13], which forces individuals to compete with other individuals that are similar to them. Other methods are domain specific, such as symbiosis [10, 19] or co-evolutionary models [24, 25]. Whichever the method, when speciation is done within a single population, a measure of similarity δ is used to group similar individuals. This can be computed in different spaces, such as: (1) fitness or objective space; (2) genotypic or decision space; and in the case of ER (3) behavioral space. In particular, behavioral space refers to the space of all the possible behaviors that a mobile robot can exhibit. Irrespective of the space in which the similarity between individuals is determined, a necessary condition is that the space should provide a multi-modal landscape for species to appear, an assumption which is often true in behavioral space for robotic tasks [22, 28].

Speciation is done in order to achieve one of two general goals. Some methods focus on finding several partial and specialized solutions. For example, the SANE [19], ESP [10] and CONE [21] methods use a symbiotic approach that co-evolves several individual neurons that are combined to construct a complete ANN. Another example is the Parisian approach, where the fitness of an individual is determined by its own *local* fitness, combined with a *global* fitness computed for a complete aggregate solution [5]. These works focus on problem decomposition and specialization, what can be called evolutionary divide and conquer [5, 27]. This contrasts with the aim of the current work that searches for a diverse set of complete and monolithic solutions, the second general goal that many speciation methods address. Considering this second goal, a speciation method searches for solutions that perform different versions of basically the same job [2]. A relevant example is [11], where evolution is used to find several alternative paths for a rigid body in 2D and 3D environments. However, that work is related with deliberate control systems, not the behavioral approach followed here. Moreover, based on the ER paradigm, the present work entails species formation for a population of ANNs, speciation within a neuroevolutionary system. Currently, one of the best approaches in this area is the Neuro-Evolution of Augmenting Topologies (NEAT) method [29], a specialized GA that evolves a population of ANNs with a variety topologies. However, the goal of speciation is to produce a functionally diverse set of solutions for a particular problem. Therefore, speciation based on topological differences should be of less interest if different species do not exhibit a difference in their functional response. Moreover, a diverse set of ANN topologies does not guarantee a diverse set of functional solutions [18]. Therefore, we argue that in ER a measure of similarity between robot controllers should focus on the actual behaviors that each controller produces, see Fig. 1.

3 Behavior-based Speciation

In order to speciate in behavioral space, we represent behaviors using *behavior signatures* expressed as character strings and determine similarity using string comparison techniques.

3.1 Behaviors and Neurocontrollers

The distinction between a behavior and an individual must be stressed because they do not represent the same concept. In EC, the genotype is given by the encoding used to represent each neurocontroller within the evolutionary process. The phenotype is the instantiation of each neurocontroller x with a specific topology. On the other hand, a behavior is a navigation strategy α induced by the ANN x within an environment $\mathcal{E}$, written as $x \overset{\mathcal{E}}{\rightsquigarrow} a$. Therefore, a behavior depends upon the phenotype of the individual and the structure of the environment. Moreover, it is assumed that each neurocontroller x induces one and only one behavior within $\mathcal{E}$. This assumption is considered true only if the initial robot heading and starting position are fixed. However, due to competing conventions a many-to-one relationship should be assumed between individuals and behaviors. Consequently, if two individuals x and y induce behaviors $x \overset{\mathcal{E}}{\rightsquigarrow} a$ and $y \overset{\mathcal{E}}{\rightsquigarrow} a$ respectively, then the notation implies that the underlying behavior α is shared by both.

It should ne noted that a behavior is considered to be a subjective concept, while a behavior signature S_α represents an objective characterization of α. It can be said that S_α is obtained by way of a behavior interpretation process, denoted by ψ. On the other hand, a behavior α can be described more comprehensively if one uses a detailed account of how the robot moves and acts within the environment. Such an approach is common in ER as well as in the field of ethology, when a researcher describes and categorizes the animal behaviors that he is attempting to understand [14]. In this respect, we will also employ a descriptive categorization of the evolved behaviors, see Section 5.1.2.

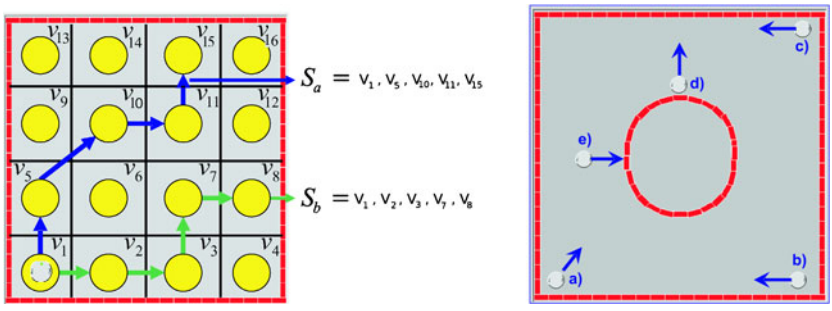
3.2 Behavior Signatures and Interpretation Process

This section defines the manner in which behaviors are described and outlines our implementation.

Definition 1 First, Let x represent an individual neurocontroller and α the behavior x induces within environment $\mathcal{E}$, written as $x \overset{\mathcal{E}}{\rightsquigarrow} \alpha$. Then, the **behavior signature** S_α represents a description of behavior a, obtained through a **behavior interpretation process** ψ, written as $\psi(\alpha) \hookrightarrow S_\alpha$.

Only a conceptual definition is given because defining each of the concepts mentioned within is not trivial. Indeed, taking measurements of specific behavior attributes, such as speed or force, is common [28]. However, the same cannot be easily done for the behavior itself. The reason for this is that ψ is an attempt to interpret a behavior as if it had concrete existence. However, because of the abstract nature of the task, there is no strict limitation on how ψ should be defined. In our

work, we employ a ψ_1 such that S_α represents the traversed path of the robot within $\mathcal{E}$. Therefore, the proposed speciation method works under the assumption that each behavior α is characterized by one and only one signature. Figure 2a gives a graphical representation of the proposed behavior signatures using ψ_1. The environment is represented using a topological map $M = (V, E)$ where V is the set of nodes in M, and E the set of connecting edges; the training environment in Fig. 2b has the same topological structure shown in Fig. 2a. The robot is 5 cm in diameter, and the world is a 1 m^2 surface, divided in a 4×4 grid with individual cells of 25 cm $\times$ 25 cm. Therefore, in each cell the robot could occupy a total of 25 individual non-overlapping sub-cells. The granularity of the grid used to define the topological map is clearly important. If the grid is too coarse then the signatures will not adequately capture behavioral differences. If the grid is too fine, then slight differences in the path generated by two similar behaviors might be overly magnified. The 4×4 seemed to provide the best trade-off in our experimental tests. A neurocontroller x, starting from an initial node $v_1 \in V$ (Fig. 2b), guides the robot across the map generating a path S, represented by the sequence of nodes visited by the robot $S = v_i, ..., v_j, ..., v_n$, see Fig. 2a. In order to obtain a signature S, a controller x navigates the robot for 4,000 cycles, which is roughly enough time for the robot to complete two laps around the environment. When the robot is moving at full speed, it requires approximately 100 cycles to cross a single cell. Therefore, the position of the robot is updated every 10 cycles, a sufficiently fine sampling rate to capture transitions within the topological map. If at a given update cycle t the node the robot occupies v^t is different from the node it occupied at the previous update cycle v^{t-1}, then v^t is added to S. However, nodes are added to S only after the robot has explored the environment during an initial stabilizing time of 500 cycles. If not for the stabilizing time, the initial characters of all signatures would be similar only because the robot is always placed in the same starting position and not due to any meaningful similarity between the behaviors. Finally, a string similarity measure $\delta(S_\alpha, S_\beta)$ can be used to compare signatures S_α and S_β because every signature is also a string.



(a) Signatures (b) The circle world

Fig. 2 (**a**) Two paths are shown, each generated with a different neurocontroller, in this case x and y. The environment is partitioned into a 4×4 graph, each node labeled from v_1 to v_{16}. If controller x induces behavior a, then the behavior signature S_a represents the traversed path of the robot, here shown in *dark lines*. (**b**) The *circle world* used for the problem of robot navigation. **a** Starting position for behavioral signatures; **b**–**e** initial positions and headings for each of the four epochs used to assign fitness

3.3 String Similarity Measures

Before introducing each measure, a few concepts need to be established. The alphabet is Σ, Σ^* is the set of strings over Σ, and $\lambda \notin \Sigma$ is the null string. Here, $\Sigma = V$, and Σ^* is the set of possible paths in $\mathcal{M}$. A string $S \in \Sigma^*$ is expressed as $S = s_1, s_2 ... s_n$, where $s_i \in \Sigma$ is the *ith* symbol of S, and $|S| = n$ the size of the string (the null string has $|\lambda| = 0$). $S^{i,j}$ is a substring of S that includes the symbols from s_i to s_j, with $1 \leq i \leq j \leq n$ and it is the null string if $i > j$. $\{S\}$ represents the set of all substrings in S, and $\{S_{a,b}\}$ the set of all substrings in two strings S_a and S_b. Furthermore, $S_a \in \Sigma^*$ can be interpreted as an ordered set S_A of characters $x \in \Sigma$. If $S_A = \{s_1, s_2 ... s_n\}$ then $K = \{1, 2...n\}$ is the ordered index set of S_A. Thus, a subsequence $\widehat{S}_a$ is a sequence of characters $s_i \in S_A$, where i is a strictly increasing sequence in the index set of K. $\{\widehat{S}_a\}$ represents the set of all $\widehat{S}_a$ of S_a.

3.3.1 N-GLD, Normalized Generalized Levenshtein Distance

The Generalized Levenshtein Distance (GLD) [32], also known as the edit distance, compares strings by various edit operations, commonly using the deletion, insertion, and substitution of symbols. If $v, u \in \Sigma$, an elementary edit operation is a pair $(v, u) \neq (\lambda, \lambda)$ written as $v \to u$, where $|v|, |u| \in \{0, 1\}$. The operations $\lambda \to v$, $v \to u$, and $u \to \lambda$, represent insertions, substitutions and deletions respectively. Then, the *edit transformation* $T_{S_a,S_b} = T_1, T_2 ... T_l$ can be defined as a sequence of edit operations that transforms S_a into S_b. Now, if γ is a weight function that assigns a non-negative real number to each edit operation T_i, such that $\gamma(T_i) \geq 0$, then the total weight of a complete edit transformation T_{S_a,S_b} is the sum of the weights assigned by γ to each edit operation in T_{S_a,S_b}, expressed as

$$\gamma\left(T_{S_a,S_b}\right) = \sum_{i=1}^{l} \gamma(T_i) \,. \tag{1}$$

Then, [32] define the GLD as

$$GLD(S_a, S_b) = \min\left\{\gamma\left(T_{S_a,S_b}\right)\right\} \,. \tag{2}$$

To account for the common situation in which $|S_a| \neq |S_b|$, a normalized version of GLD is required [32], defined for two strings $S_a, S_b \in \Sigma^*$ as

$$\delta_{N-GLD}(S_a, S_b) = \frac{2 \cdot GLD(S_a, S_b)}{\alpha(|S_a| + |S_b|) + GLD(S_a, S_b)} \,, \tag{3}$$

where $\alpha = \max\{\gamma(v \to \lambda), \gamma(\lambda \to u),\ v, u \in \Sigma\}$, and $\delta_{N-GLD}(\lambda, \lambda) = 0$. Finally, following [32] the weight function γ is defined as $\gamma(v, v) = 0$, $\gamma(v, u) = 1$, and $\gamma(v, \lambda) = \gamma(\lambda, v) = 1$ for any $v, u \in \Sigma$.

3.3.2 LCD, Linguistic Complexity Distance

The linguistic complexity of a string S is the ratio of the number of substrings of S to the maximum number of substrings that can be obtained from a string of the same

length on the same alphabet [15]. Using the concept of linguistic complexity, [15] defines a Linguistic Complexity Distance δ_{LCD} between two strings as

$$\delta_{LCD}(S_a, S_b) = 2 \cdot \frac{|\{S_{a,b}|\}}{|\{S_a\}| + |\{S_b\}|} - 1 \,. \tag{4}$$

3.3.3 LSS, Largest Subsequence Similarity

Using the concept of subsequence, two strings $S_a, S_b \in \Sigma^*$ can be compared using the following Largest Subsequence Similarity δ_{LSS},

$$\delta_{LSS}(S_a, S_b) = 1 - \frac{\max\{|\widehat{S_b}|,\ \widehat{S_b} \in \{\widehat{S_a}\}\}}{|S_a|} \,. \tag{5}$$

3.4 Behavior Grouping and Speciation Evaluation

Before presenting the experimental setup, three new concepts are introduced: *species behaviors*, *singular behaviors*, and the *behavior speciation ratio*. These concepts establish a conceptual framework that will allow us to discuss the experimental results obtained with the proposed approach.

3.4.1 Species Behaviors

First, we define the set of behaviors that serve as representatives for each species.

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

$$\mathcal{P} = \bigcup_{k=1}^{M} R_k \ \textit{where}\ R_k \cap R_l = \varnothing \ \textit{for}\ k \neq l \,. \tag{6}$$

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 $\mathcal{B} = \{\alpha^1, ...\alpha^i, ...\alpha^L\}$ of L behaviors, such that $\forall\ \alpha^i \in \mathcal{B}$ if $x \overset{\mathcal{E}}{\rightsquigarrow} \alpha^i$ and $x \in R_k$ then

$$\begin{aligned} f(x) &> \sup\{f(y) |\ \forall\ y \in R_k,\ y \neq x\} \\ &\wedge\ f(x) > h \,, \end{aligned} \tag{7}$$

where h is called the *behavior threshold*.

Therefore, every $\alpha^i \in \mathcal{B}$ is induced by one and only one neurocontroller $x \in \mathcal{P}$, and every such neurocontroller is the super-individual of its species. The set $\widehat{\mathcal{B}}$ of all such x is called the set of **species neurocontrollers** and its relationship with $\mathcal{B}$ is written $\widehat{\mathcal{B}} \overset{\mathcal{E}}{\rightsquigarrow} \mathcal{B}$. The *species behaviors* are contingent on the environment $\mathcal{E}$ that the neurocontrollers interact with, the training environment. 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 unique navigation strategies. Finally, as noted in Definition 2, the inclusion of a behavior into $\mathcal{B}$ depends on the behavior threshold h, which is set empirically rather than being derived, or chosen, a priori.

3.4.2 Singular Behavior

We now define the set of structurally unique behaviors.

Definition 3 The underlying set $\mathcal{I}$ of multiset $\mathcal{B}$, is called the set of **singular behaviors**. Every behavior $\alpha \in \mathcal{I}$ exhibits a unique navigation strategy within environment $\mathcal{E}$.

Insofar as a perfect grouping - or species formation - of individuals is expected, a distinction between $\mathcal{I}$ and $\mathcal{B}$ would be needless. However, a distinction is necessary because the speciation mechanism can produce errors. These errors can occur because the speciation process works under the following assumptions: (1) each neurocontroller x induces one and only one behavior α within $\mathcal{E}$; and (2) each behavior α can be instantiated by one and only one signature S_α.

The first assumption will not hold when a neurocontroller is not robust. In such a case, the errors in sensor readings and actuator responses can cause the robot to behave differently when confronted with the same situation. The second assumption relates to how behavior signatures are defined, and the manner in which comparisons are made. The signatures we propose, based on the path followed by the robot, provide an approximate description of a behavior. However, such a description can sometimes fail to capture the finer details that make two behaviors similar or dissimilar. For instance, two robots can follow the same path but the manner in which each robot turns when faced with an obstacle might be different. In such a case, the path cannot capture such subtle traits.

Returning to the concept of singular behaviors, it is important to understand that the goal of speciation is to produce a large set $\mathcal{I}$ because this indicates that many unique solutions have been found. However, defining membership to $\mathcal{I}$ requires a method that determines the *uniqueness* of each behavior. Therefore, to avoid misclassification another behavior interpretation process ψ_2 is proposed; in this case, it is carried out by a human expert using distal information. For ψ_2, behavioral traits are visually identified from the path generated by x and used as comparative criteria. The proposal is to use a descriptive account of each behavior, an approach that is indeed subjective but also consistent with ER research [22] and with interactive evolution [12]. In sum, a behavior signature given by $\psi_1(\alpha) \hookrightarrow S_{\alpha,1}$ only represents an instance of a given behavior α. S_α describes the path the robot followed, caused not only by the underlying controller but also by the interactions between the sensors and the environment during a given time interval. Conversely, the actual behavior α is more comprehensively described by the set of behavioral traits it exhibits, captured by $\psi_2(\alpha) \hookrightarrow S_{\alpha,2}$.

3.4.3 Behavior Speciation Ratio

Finally, in order to estimate the performance of the speciation method a numeric measure is proposed, the behavior speciation ratio.

Definition 4 Given the set of *singular behaviors* $\mathcal{I}$ and the multiset of *species behaviors* $\mathcal{B}$, the **behavior speciation index** $\mathcal{BSR}$ is given by

$$\mathcal{BSR} = \frac{|\mathcal{I}|}{|\mathcal{B}|} \,. \tag{8}$$

The $\mathcal{BSR}$ characterizes the ability of a speciating algorithm to generate unique behaviors within each species. When $\mathcal{BSR} \approx \frac{1}{|\mathcal{B}|}$, the species primarily converge to the same behavior. Conversely, when $\mathcal{BSR} \approx 1$ the super individual of each species is unique within $\mathcal{B}$, hence $\mathcal{B}$ would be a set and not a multiset. This constitutes a desirable outcome because it indicates that the algorithm is correctly grouping individuals based on behavioral similarities. A method that consistently produces a $\mathcal{BSR}$ close to 1 also has a practical use, because one could take the species behaviors and assume that all of them are unique, the ideal speciation for ER.

To evaluate our behavior-based speciation two stages are used. The first is on-line, when a multi-set of species behaviors $\mathcal{B}$ is obtained using ψ_1. The second stage is off-line, when a human observer employs ψ_2 to identify the set of singular behaviors $\mathcal{I}$ and compute the $\mathcal{BSR}$.

4 Implementation and Experimental Setup

4.1 The ER System for Behavior-based Speciation

Figure 3 is a high-level view of the proposed algorithm. Our proposed speciation strategy does not depend upon a single type of evolutionary system, and could be incorporated into different methods. Here, we use the basic NEAT algorithm as the basis for our evolutionary system [29].[1] Stanley and Miikkulainen [29]. NEAT is a generational GA with fitness proportional selection that uses a variable-size representation that allows it to evolve ANNs of different sizes and topologies. The algorithm can progressively generate larger networks after being initialized with a population of networks that share the same minimal topology. NEAT can produce a variety of network topologies through the use of speciation based on topological similarity. ANNs are compared based on the number of disjoint genes D and excess genes G between them (genes represent individual neuron), as well as differences between the connection weights in links that both networks have. Hence, the similarity measure used by NEAT is given by

$$\delta_{NEAT} = \frac{c_1 \cdot G + c_2 \cdot D}{N} + c_3 \cdot \overline{W}\,, \tag{9}$$

where $\overline{W}$ is the average weight difference of matching genes, N is the number of genes in the larger genome, and c_x are weight coefficients set to $c_1 = c_2 = 1$ and $c_3 = 0.4$. Thus, given a similarity threshold δ_t a new individual a is added to the first species B where its distance δ_{NEAT} to a randomly selected species member $b \in B$ is $\delta_{NEAT}(a, b) < \delta_t$. If no species is found then a new species A is created for a. Finally, NEAT uses fitness sharing to promote diversity [9].

In our work, the measure of topological similarity is substituted with the string similarity measures of Section 3.3. Thus, species are formed based on the behavioral outcome of each neurocontroller instead of the topology of each neural network.

[1] Source code downloaded from the Neural Networks Research Group of the University of Texas at Austin: http://www.cs.utexas.edu/nn/.

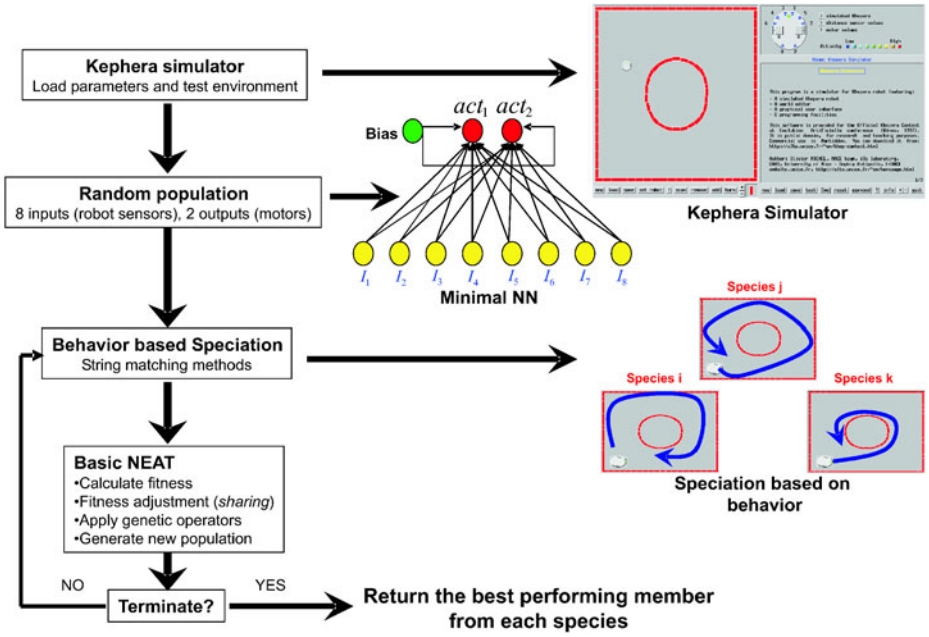
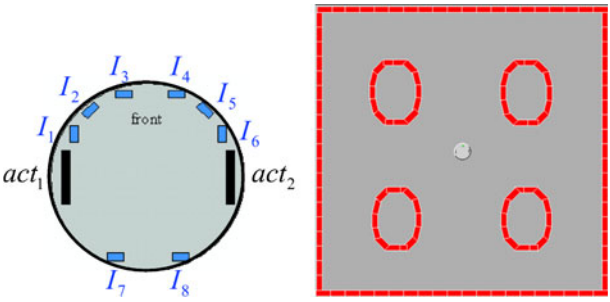


Fig. 3 An overview of the ER system for behavior-based speciation. The Khepera simulator loads all the algorithm parameters and provides the user interface. The initial population is created with the minimal topology for the neurocotrollers. During evolution, the process of behavior-based speciation 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 *Species Behaviors* $\mathcal{B}$

Behavior signatures are obtained for each ANN by placing the robot at position $V1$ with a 45° heading, depicted in Fig. 2a, b. 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 (act_1 and act_2) with randomly assigned weights, see Fig. 3. Note that even though species formation does not consider the topology of each network, instead focusing on behavioral similarities, the underlying representation used by NEAT is still able to generate new network topologies through crossover and mutation. The evolutionary algorithm is integrated into the Khepera Simulator where the robot parameters and training environment are loaded, as shown in Fig. 3.

4.2 The Khepera Robot and Simulator

The Khepera is a well known robot, very common within ER research [22]. It has two DC motors that work as actuators, act_1 and act_2, eight infrared proximity sensors $I_1, I_2, ..., I_8$, and eight light sensors $L_1, L_2, ..., L_8$; the robot is illustrated in Fig. 4a. The Khepera has a simple architecture which makes it ideal for ER. Nevertheless, using a real robot to evolve neurocontrollers can be quite slow and cumbersome [22]. Therefore, in our work we use the freeware Khepera Simulator version 2.0 [16].



(a) Khepera (b) The 4-circle world

Fig. 4 **a** The Khepera robot, with eight proximity sensors and two motor actuators. **b** The 4-circle world, used as the second training environment for the navigation problem

4.3 Autonomous Navigation

We test the performance of our approach using two problems of robot navigation. Each problem is clearly multi-modal in behavioral space, a necessary condition to test our proposal. For these problems we use the Khepera robot, described next.

4.3.1 Training Environments

The proposed behavior-based speciation is tested on two navigation problems using different training environments.

The Circle World The first training environment is similar to the one used in [17], see Fig. 2b. The environment is simple, a square room with a large obstacle in the middle, around which the robot must navigate. Despite its simplicity, the environment offers a multi-modal landscape in behavioral space.

The 4-circle World The second environment contains four equally spaced circular obstacles, its depicted in Fig. 4b. For this environment the navigation problem can also be solved in several different ways. Indeed, this environment is more complex, and the robot could use more navigation strategies in order to explore it.

4.3.2 Fitness Evaluation

The type of behavior that the EA 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 [17], for each neurocontroller x,

$$f(x) = \frac{1}{N \cdot M} \sum_{j=1}^{M} \sum_{k=1}^{N} V_k \left(1 - \sqrt{\triangle v_k}\right) (1 - \varphi_k) \,, \tag{10}$$

In the above equation, property (1) is promoted by V_k. which is the sum of the two motor speeds at time step k. Likewise, property (2) is promoted by the absolute difference between the two motors $\sqrt{\triangle v_k}$. Finally, φ is the normalized activation

value of the infrared sensor with the highest activation, this term promotes property (3). Therefore, fitness is maximized with better performance. Moreover, M is the number of test runs, or epochs, and N the total number of time steps or cycles within the environment during an epoch j. The number of epochs is $M = 4$, with the initial position and heading of the robot for each epoch are shown in Fig. 2b.

Furthermore, we want to obtain comparative results for each similarity measure. Here, we compare the string distance measures N-GLD, LCD, LSS, and the topological measure used by NEAT. Moreover, a canonical GA with a fixed topology ANN (the minimal topology possible) is also tested. The GA is obtained by setting the similarity threshold $\delta_t = 0$, and not allowing new synapses or nodes to be added. The parameters employed by each method are shown in Table 1; all the methods share most of the runtime parameters except for δ_t. In the case of NEAT, this parameter was chosen as indicated by [29], and it was set experimentally for the other similarity measures. In each case, the goal was to obtain a steady number of species during evolution.

4.4 Homing Navigation with Battery Recharge

The second set of experiments addresses the problem of homing navigation with internal robot dynamics, based on [8] where the problem is posed as follows. The robot is equipped with a limited but rechargeable energy source, and it must navigate without collisions within the environment for as long as possible. The environment includes an area where the battery of the robot can be recharged and this area is marked by a unique environmental feature, such as a light source. The robot must learn a navigation strategy that allows it to navigate while periodically recharging its battery. In order to accomplish this goal, the control system can monitor the energy level of the battery, and it can monitor the readings of additional sensors that allow it to identify the recharge area.

We implement two versions of the homing problem. The first one uses a Khepera and an ANN with 20 neurons: eight proximity sensors, eight light sensors, one battery sensor, one bias, and two outputs. The second version is implemented on a simulated and real Pioneer P2-AT robot, a larger and more complex robot.

Table 1 The parameters used by the ER system

Name	Value
Runs	6
Population	100
Generations	50
Add synapse probability	0.1
Add node probability	0.03
Interspecies mating rate	0.05
Crossover rate	0.75
Compatibility threshold	$\delta_{N-GLD} = 0.4$, $\delta_{LCD} = 0.85$, $\delta_{LSS} = 0.6$, $\delta_{NEAT} = 3$.
Transfer function	Sigmoidal
Behavior threshold	$h = 3.7$

4.4.1 Pioneer P2-AT Robot

The Pioneer P2-AT (ActivMedia Robotics) is a four-wheeled robot equipped with a digital camera and a sonar belt with twelve sensors, see Fig. 5c. It is necessary to model the Pioneer robot as a Braitenberg vehicle, see Fig. 5. Unlike the Khepera, however, the Pioneer has four wheels, and instead of directly controlling each motor separately the software interface provides commands that allow us to control the robots movements. In this work, only three of these commands are sufficient to move the robot; these are: *speed*, *move*, and *turn*. Therefore, the output from each neurocontroller provides the input values for each command, simulating the control architecture of Fig. 5b.

4.4.2 Training Environments

Two different environments are used, one for each robot.

Khepera Robot For the Khepera, the environment does not have an obstacle and the recharge area is located in the upper-right corner illuminated by three light sources, see Fig. 6a. It uses the same starting positions for each epoch, and the robot is placed in the center facing the recharge area when the signature is generated.

Pioneer P2-AT The homing activity was slightly modified, instead of using light, the robot detects the recharge area with the on-board camera and an object detection algorithm that detects faces [31] that are affixed to the slanted wall in the recharge area, see Fig. 6b. During simulation, we know the position and orientation of the robot, as well as the cameras field of view, which is 48.8°. Hence, we can estimate the area of the surrounding walls that the camera is able to observe at any time. In this way it is possible to predict when the robot will detect a face on the wall. However, two conditions should be fulfilled in order to recharge the battery: (1) the robot should be within the recharge area; and (2) the robot should be oriented towards the walls with the face images. The robot can determine when it is within the recharge area by using the size of the bounding-box around a detected face. The initial ANN topology contains 14 input nodes (twelve for sonars, one for battery level and one for the visual module), one bias and three output neurons.

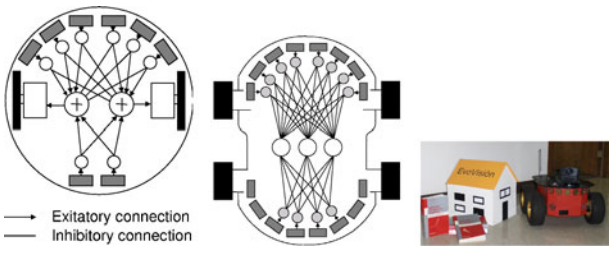
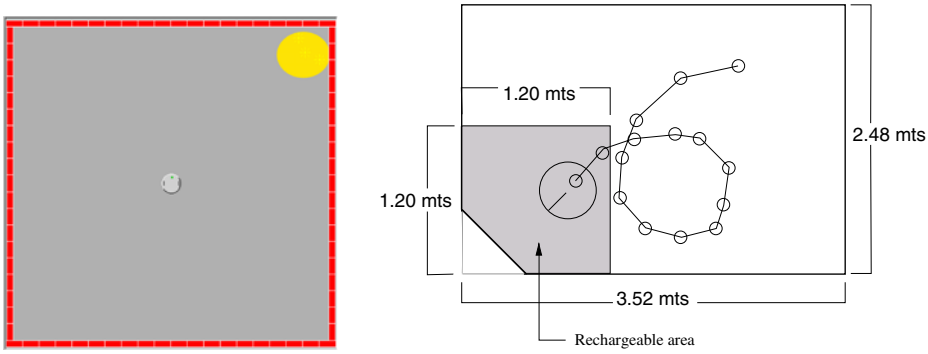


Fig. 5 **a** Braitenberg vehicle. **b** Conceptual Pioneer P2-AT. **c** Pioneer P2-AT robot with camera



(a) Khepera robot (b) Pioneer P2-AT

Fig. 6 The training environment for the homing navigation problem: **a** Shows the light source in the *upper right corner* and the Khepera robot in the *center*; **b** shows the Pioneer robot within the recharge area in the *bottom left corner*

4.4.3 Fitness Evaluation

For the homing problem with battery recharge, the fitness function was simplified, following [8, 22]. What is of interest in this case is to observe if the evolutionary algorithm can produce robot behaviors that find and take advantage of the recharge area. Therefore, the fitness for an individual neurocontroller x is given by,

$$f(x) = \frac{1}{N \cdot M} \sum_{j=1}^{M} \sum_{k=1}^{N} V(1 - \varphi_k) \,, \tag{11}$$

where V_k is the sum of the two motor speeds at time step k, φ is the normalized activation value of the proximity sensor with the highest level of activation, M is the number of epochs, and N the number of cycles within the environment during an epoch j. The fitness function does not explicitly measure the desired homing behavior, however by summing over the total number of cycles it encourages the robot to move within the environment for as long as possible. Moreover, the battery only provides enough simulated energy for the robot to navigate for 1,500 cycles, and each epoch lasts 4,500 cycles. Hence, if the robot wants to navigate within the environment for the maximum alloted time it must recharge its battery only when needed because it is allowed to recharge only three times during each epoch, see [8, 22] for more details.

5 Experimental Results

5.1 Autonomous Navigation: Circle World

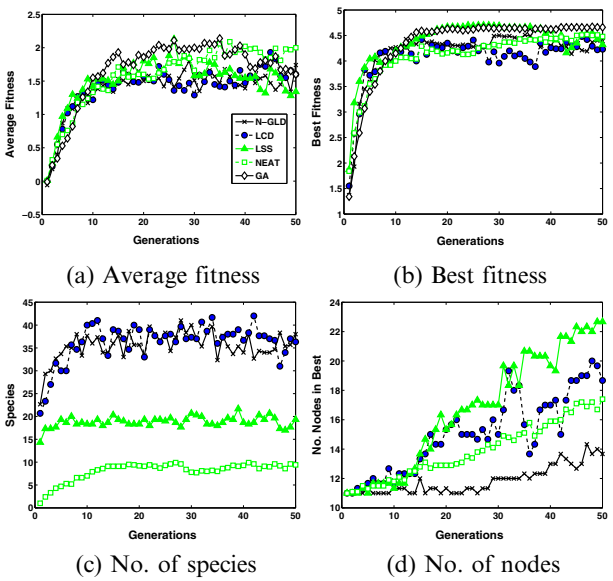
This subsection presents the experimental results for the navigation problem using the circle world, Fig. 2b.

5.1.1 Evolution Statistics

Figure 7a–d presents several comparative plots. All graphs are plotted relative to the number of generations, and represent averages over the total number of runs. Figure 7a presents a plot for the average population fitness. Similar performance is exhibited by all measures, with the GA achieving a slightly higher average fitness. This is an expected result because the GA lacks a diversity preservation mechanism, thus allowing the population to converge towards a single fitness extrema. Figure 7b shows the fitness of the best individual in each population. All methods converge towards similar fitness, with the GA and the LSS method achieving a higher average performance. For the GA this can be anticipated because the algorithm exploits the best individuals at each generation; however, the higher performance of LSS was unexpected. Figure 7c shows a plot for the total number of species in the population; this graph excludes the GA. The N-GLD and LCD produce a larger number of species, while LSS and NEAT are substantially more compact. This observation suggests that species formed through the N-GLD and LCD measures are less stable, and the grouping of individuals is more difficult. Figure 7d shows the total number of nodes in the super-individual of each population. The plot describes the amount of complexity that each measure produces, it reveals that all methods stay between a similar range of values, with a minimum of 11 nodes and a maximum of 22 nodes, except for N-GLD that only reaches up to 14 nodes. All follow a similar monotonic and steady increase, except for LCD.

5.1.2 Behavioral Traits and Categorization

Section 3 introduced the concepts of species behaviors $\mathcal{B}$ and singular behaviors $\mathcal{I}$ for an environment $\mathcal{E}$. Even though the multiset $\mathcal{B}$ can be automatically obtained from $\mathcal{P}$,



(a) Average fitness (b) Best fitness

(c) No. of species (d) No. of nodes

Fig. 7 Performance plots that compare all four similarity measures as well as a simple GA without speciation, using the *circle world* as the training environment. All plots share the same legend, however **c** and **d** do not include results for the simple GA

defining membership for $\mathcal{I}$ requires off-line evaluation of the behaviors. Therefore, in order to decide which behaviors belong in $\mathcal{I}$, a special list of behavioral traits are defined and used for categorization, these are: *Direction*, *Turning*, *Navigation*, *Looping* and *Circling*; see Table 2. A graphical representation is presented in Fig. 8a, where sample behaviors depict each of the traits. The values that each trait may take are self-explanatory. These traits were identified from a careful analysis of the behaviors obtained from different runs. The selection of the traits was done by the authors, and the assignment of values to a particular behavior was done by one observer and independently confirmed by another. In all the cases presented here, both observers assigned the same trait values.

With the set of traits in Table 2, depicted in Fig. 8a, we can compare behaviors. When two behaviors $\alpha, \beta \in \mathcal{B}$ differ in the value of one of the five traits then they are said to be different. Figure 8b presents a sample comparison of two behaviors using all of the traits. With these traits it is possible to determine the underlying set of singular behaviors contained within the multiset of species behaviors.

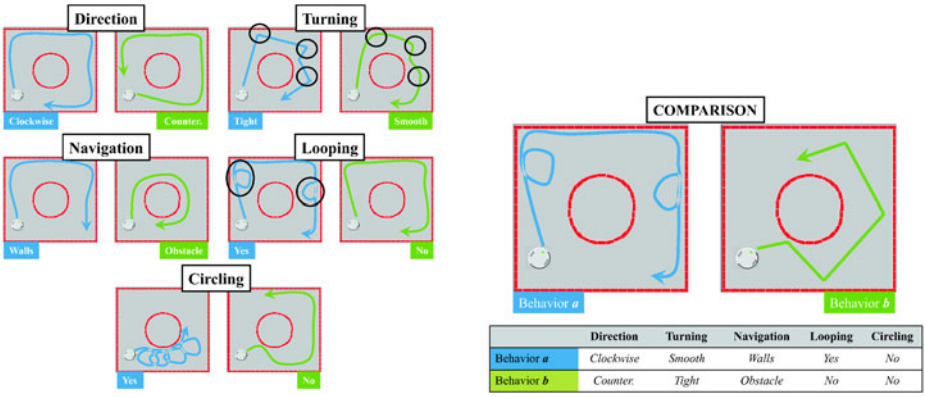
The traits we have chosen appeared as clear and repetitive patterns within the behaviors induced by more than a hundred evolved neurocontrollers. These traits allow us to construct a descriptive account of the structure that each behavior exhibits. However, we do not claim that these traits give a precise or exhaustive description, nor do we claim that they are optimal. Nevertheless, we do believe they are a useful tool to compare and categorize behaviors based on practices used in ethological research [14].

5.1.3 Testing Environment

Besides the training environment another environment is used for testing, see Fig. 9. Note the marked difference between the training and testing environments; this makes navigation in the latter a difficult task for a controller evolved in the former. The test environment has three *trapping regions*, which present difficult scenarios for the control system from which the robot cannot escape. Therefore, good navigation within the testing environment implies that the robot can explore the environment without getting trapped.

Table 2 Behavioral traits used for behavior categorization

Trait	Description	Values
Direction	The main direction a behavior follows while traversing the environment.	Clockwise–Counterclockwise
Turning	When confronted with an obstacle some behaviors prefer tight turns by stopping and turning in place; others perform a smoother turn while the robot is in motion; others use both.	Tight–Smooth–Both
Navigation	Some behaviors use the outer walls to navigate; others use the center obstacle; and some have no preference.	Walls–Obstacle–None
Looping	In order to reposition the robot and continue on a primarily straight line, some behaviors use a distinctive looping motion.	Yes–No
Circling	Not all high performance behaviors favor straight line movement, some prefer a circling type of motion.	Yes–No



(a) Behavioral Traits (b) Comparison

Fig. 8 **a** Behavioral traits: *Direction*, *Turning*, *Navigation*, *Looping* and *Circling*. Each is shown with a behavior that exhibits the values that the trait may take. For *Turning*, the **both** value is assigned to a behavior exhibiting both types of turns; similarly for the **none** value of the *Navigation* trait. **b** Different behaviors can be compared using their behavioral traits. Behaviors α and β are clearly different, and these differences are expressed using the set of proposed traits

5.1.4 Behavior-based Speciation

We now proceed to a qualitative evaluation of the behaviors obtained by each similarity measure. A total of six runs of each method, produced six corresponding $\mathcal{B}$. Here, only one set of species behaviors $\mathcal{B}$ from each method is presented. Figure 10a–c shows five species behaviors obtained with each of the similarity measures: N-GLD, LCD, LSS and NEAT. The species behaviors we present are those with the highest fitness values. The top row of each figure shows the behavior within the training environment. Because these behaviors are taken from $\mathcal{B}$, a color coding scheme is used to identify which of them correspond to the same singular behavior in $\mathcal{I}$. When two behaviors are depicted with the same color they represent two instances of the same singular behavior. Next, a table lists: (1) the behavioral traits of each *species behavior*; (2) the associated fitness value for the *species neurocontroller*; and (3) if it can adapt to the testing environment. The traversed path within the testing environment is shown in the bottom row of each figure. The neurocontroller that induces behavior α is called x^{α}. From these results, we can state the following. For N-GLD, the species behaviors shown are unique; hence, they are singular behaviors. Nonetheless, most neurocontrollers perform poorly in the test environment. In the

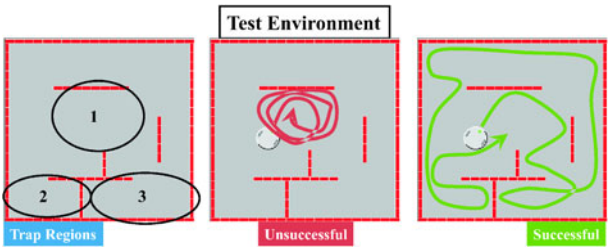


Fig. 9 The *Test Environment*, from *left to right*: **1** The regions within the test environment where a robot controller can get **trapped**; **2** a sample behavior getting trapped in *region 1*; and **3** a behavior that successfully navigates without getting trapped

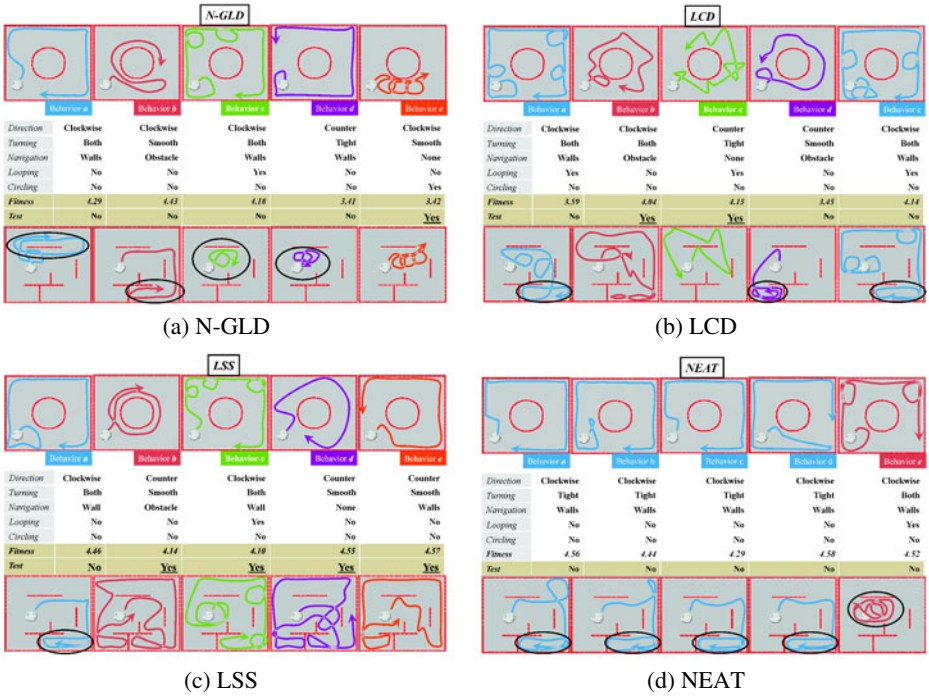


Fig. 10 (**a**) Five *species behaviors* obtained with the N-GLD measure. All five behaviors are also *singular behaviors*. Only x^e is able to avoid getting *trapped* in the test environment. (**b**) Five *species behaviors* obtained with LCD. Here, behavior **a** and **e** are instances of the same singular behavior; hence, only four behaviors in $\mathcal{I}$ are shown. Furthermore, the similarity between **a** and **e** is likewise manifested by the unsuccessful navigation within the testing environment of x^a and x^e. (**c**) Five *species behaviors* obtained with LSS. All five behaviors are also *singular behaviors*. Furthermore, they also represent solutions capable of adapting their traits in order to successfully navigate within the testing environment; except for x^a. (**d**) Five *species behaviors* obtained with NEAT. Observe how four of the behaviors represent the same navigation strategy. Therefore, speciation done in topological space does not seem to be able to produce a functionally different set of species. Additionally, solutions are overfitted to the training environment, evidenced by their unsuccessful navigation within the test environment

case of LCD, only behaviors **a** and **e** are not unique; thus, four singular behaviors were identified, and only neurocontrollers x^b and x^c were able to navigate within the testing environment. Then, for LSS every species behavior is also a singular behavior. Furthermore, four of the five neurocontrollers were able to explore the test environment without getting trapped. Finally, the results for NEAT show that differences in topological space do not imply behavioral differences. Four of the five species behaviors exhibit the same basic strategy, hence only two singular behaviors were found. Furthermore, controllers were overfitted to the training environment and performed poorly within the test environment.

Based on the above results, we conclude that the main objective of our work was achieved. This is evidenced by the total number of singular behaviors produced by each method, from which it follows that the algorithm was in fact able to produce various neurocontrollers that perform different versions of basically the same job.

From a detailed analysis of all the experimental runs, the results presented in Fig. 10 are considered to be representative for each similarity measure. Nevertheless, these results were all produced in a single run of the algorithm, the statistical significance of these results is not reliable. Therefore, a statistical t-test is performed to obtain 95% confidence intervals, $\mu - \tau < \mu < \mu + \tau$, regarding the mean cardinality of $\mathcal{B}$ and $\mathcal{I}$; Table 3 summarizes the results.

Speciation with the N-GLD and LCD measures produces a higher average of species behaviors. However, the null hypothesis is only rejected between N-GLD and NEAT regarding the cardinality of $\mathcal{B}$. Moreover, both N-GLD and LCD generate more than twice as many species as does LSS and NEAT, see Fig. 7. Hence, an overwhelming majority of those species are in lower fitness areas of the search space, while the opposite is true for LSS and NEAT. With regards to the cardinality of $\mathcal{I}$, the null hypothesis is only rejected between all the string similarity measures (N-GLD, LCD and LSS) and NEAT, while no significant difference is found among the three string similarity measures. Furthermore, based on the small amount of singular behaviors found by NEAT, it can be stated that most of the species found through topological speciation converged towards the same navigation strategy. On the other hand, behavior-based speciation performs a better exploration of behavioral space as indicated by the $\mathcal{BSR}$. However, the statistical evidence does not suggest that any of the string measures is superior than the rest. Nevertheless, the results do show that LSS produces more robust controllers that are able to adapt to environmental changes. In this respect, LSS outperforms all other similarity measures. Moreover, for LSS the $\mathcal{BSR}$ almost reaches the ideal value of 1.

One justification in favor of behavior-based speciation is the fact that behavioral differences do not depend upon topological dissimilarities between the neurocontrollers. In order to test this argument the species behaviors presented above are compared based on their ANN topology. In our work, each ANN is represented using the NEAT chromosome, where genes encode the input and output node of each synapse, the minimal topological elements. Moreover, the genes in one ANN can be identified in another ANN using historical markings which track the appearance of a specific gene within the population [29]. Also, each gene can be either enabled or disabled, in the former case the gene is expressed in the ANN. Therefore, a topological comparison can be obtained by computing the percentage of enabled genes that appear in one ANN and that are also present and enabled in another. The results are shown in Table 4a, b and c, for each of the species behaviors found with each of the string similarity measures, N-GLD, LCD and LSS, respectively. For instance, in the case of the LSS measure (see, Table 4a) all of the genes in x^d are also present in x^e, thus there is a 0% difference between the ANNs. Moreover, there is

Table 3 Statistical comparison for the navigation problem in the circle world

Measure	$\lvert\mathcal{B}\rvert$			$\lvert\mathcal{I}\rvert$			$\mathcal{BSR}$
	μ	σ	$\pm\tau$	μ	σ	$\pm\tau$	
N-GLD	**6.33**	1.36	1.12	3.33	0.51	0.42	0.52
LCD	5.33	1.86	1.53	3.33	0.51	0.42	0.62
LSS	4.33	1.36	1.12	**3.66**	1.05	0.85	**0.84**
NEAT	5	1.49	1.22	1.90	0.56	0.46	0.38

The table shows the cardinality of $\mathcal{B}$ and $\mathcal{I}$, as well the average $\mathcal{BSR}$; bold indicates best results

Table 4 Topological comparison between species behaviors with N-GLD (a), LCD (b) and LSS (c)

	x^a	x^b	x^c	x^d	x^e
(a) N-GLD					
x^a(size 39 genes)	0%	2.5%	2.56%	28%	23%
x^b (42 genes)	7.1%	0%	7.1%	31%	26%
x^c(39 genes)	2.5%	2.5%	0%	28%	23%
x^d(36 genes)	22%	22%	22%	0%	2.7%
x^e(40 genes)	25%	25%	25%	12%	0%
(b) LCD					
x^a(size 55 genes)	0%	9%	3.6%	9%	9%
x^b (55 genes)	9%	0%	3.6%	3.6%	7.2%
x^c(58 genes)	8.6%	7%	0%	7%	10%
x^d(54 genes)	7.4%	1.8%	1.8%	0%	5.5%
x^e(68 genes)	26%	25%	25%	25%	0%
(c) LSS					
x^a(size 64 genes)	0%	3.1%	0%	3.1%	3.1%
x^b (67 genes)	7.5%	0%	7.5%	1.5%	1.5%
x^c(64 genes)	0%	3.1%	0%	3.1%	3.1%
x^d(66 genes)	6%	0%	6%	0%	0%
x^e(67 genes)	7.5%	1.5%	7.5%	1.5%	0%

only a 1.49% difference between x^d and x^e. Therefore, we can affirm that x^d and x^e are topologically very similar. However, the behavior induced by each controller is quite different, this is shown in Fig. 10 above. In fact, the ANNs found with the LSS measure share a similar topological structure without sacrificing behavioral diversity; on the contrary, the diversity is enhanced. Behavior-based speciation, especially with the LSS measure, produced ANNs which are topologically very similar, and are still quite different with respect to the behaviors they induce.

5.2 Autonomous Navigation: 4-Circle World

This subsection presents the experimental results for the navigation problem using the 4-circle world shown in Fig. 4b. In this instance, we only compare the LSS measure with the NEAT method due to the promising performance exhibited by the latter in the previous problem. All experiments were executed using the same parameters presented in Table 1, except that the total number of runs was set to five.

5.2.1 Evolution Statistics

Figure 11a–d present comparative plots between the LSS measure and NEAT. All graphs are plotted relative to the number of generations and represent the average over the total number of runs. In Fig. 11a we present the average population fitness, and in Fig. 11b we present the fitness of the best individual solution. In both cases the performance is quite similar based on average performance. In Fig. 11c we plot the total number of species generated by each method. In this comparison, NEAT generates more species than the LSS measure. The NEAT method shows a slight monotonic increase even in the final generations, generating over 20 species. On the other hand, the LSS measure reaches an asymptotic upper bound of around 15 species early in the evolutionary process, this behavior is consistent with the results

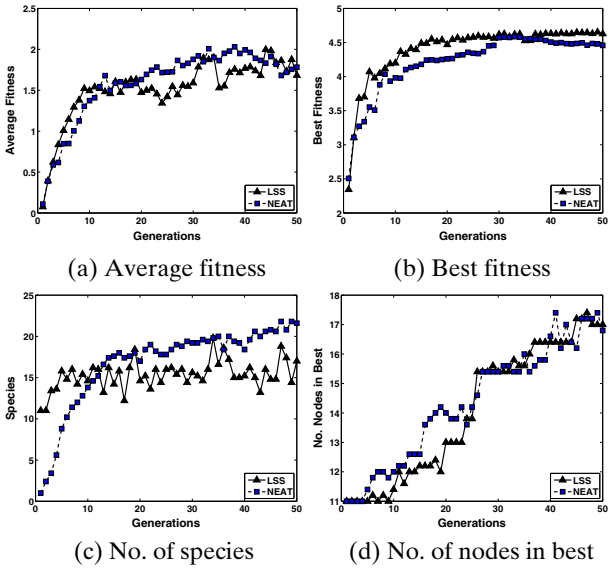


Fig. 11 Performance plots that compare behavior-based speciation with the LSS measure and the NEAT method on the 4-circle world navigation problem. All plots are averages over the total of five independent runs of the algorithm

obtained in the previous training environment, see Fig. 7c. Finally, Fig. 11d plots the total number of nodes in the best solution within the population; both measures produce ANNs of similar sizes.

5.2.2 Behavior-based Speciation

As stated above, each algorithm was executed a total of 5 times, and each run produced a corresponding set of species behaviors $\mathcal{B}$. Then, using the same behavioral traits from Table 2, we identified the corresponding set of singular behaviors $\mathcal{I}$ for each run, Table 5 summarizes the results. The table shows the average and standard deviation for the size of each set and the average $\mathcal{BSR}$ computed for each method. For this experiment, the number of singular behaviors found by each method is very similar, given the average and standard deviation. However, a one-sided t-test using a 99% confidence interval shows that the average $\mathcal{BSR}$ for the LSS measure is indeed larger than the one computed for the NEAT method. This result indicates that the LSS measure achieves a better grouping of neurocontrollers based on the behaviors they induce, with only a small overlap among different species in behavioral space. Conversely, the NEAT method tends to generate a much less efficient categorization of behaviors, something that should be expected given that it does not explicitly analyze the behaviors themselves.

Table 5 Statistical comparison of LSS and NEAT for the navigation problem in the 4-circle world

Measure	$\lvert\mathcal{B}\rvert$		$\lvert\mathcal{I}\rvert$		$\mathcal{BSR}$
	μ	σ	μ	σ	
LSS	5.4	2.60	4.2	1.64	0.82
NEAT	7.4	2.07	3.8	0.83	0.52

However, it is possible to argue that because NEAT produces an almost equivalent number of singular behaviors, then no real difference among both methods exists. Nevertheless, we believe that the diversity of behaviors found by the NEAT method is a product of the multi-modal nature of the problem itself and not due to any fundamental property of the topology-based speciation. On the other hand, the LSS measure produces a much better grouping of neurocontrollers using the concept of behavioral space. We make these claims based on the following observation.

For the problem of robot navigation the performance of the LSS measure in both environments is basically equivalent, see Tables 3 and 5. The number of species and singular behaviors, $|\mathcal{B}|$ and $|\mathcal{I}|$, are greater for the 4-circle world than for the simpler circle world. Indeed, this result was expected because the former environment is more complex and multi-modal than the latter. However, the $\mathcal{BSR}$ is basically the same in both instances, hence the LSS measure is capable of correctly categorizing behaviors in a manner which is independent of the training environment that is used. On the other hand, the NEAT method is more sensitive to the characteristics of the environment in which evolution is carried out. Indeed, there is a significant increase in the size of $\mathcal{I}$ for the 4-circle world when compared with the simpler environment. However, the $\mathcal{BSR}$ is still significantly lower for NEAT when compared with LSS, a result which reaffirms that topological diversity does not guarantee behavioral diversity. Hence, we propose that the number of singular behaviors produced by NEAT in this case is an artifact of the training environment because no substantial increase in $\mathcal{BSR}$ was obtained.

Finally, in Fig. 12 we show singular behaviors obtained with the LSS measure, each row corresponds to a set of results obtained in different runs. The diversity of behaviors produced within this environment is much richer than the diversity for the simpler circle world, the LSS measure indeed produces several distinct and unique navigation strategies.

5.3 Homing Navigation: Khepera Robot

Here we present the experimental results for the homing problem with battery recharge and the Khepera. For this problem we compare the LSS measure with the basic NEAT method, and all experiments were executed using the same parameters

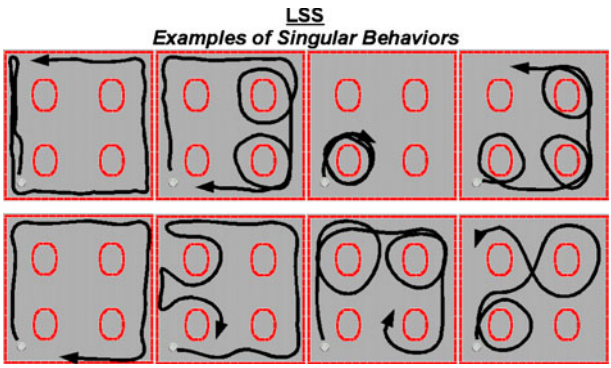
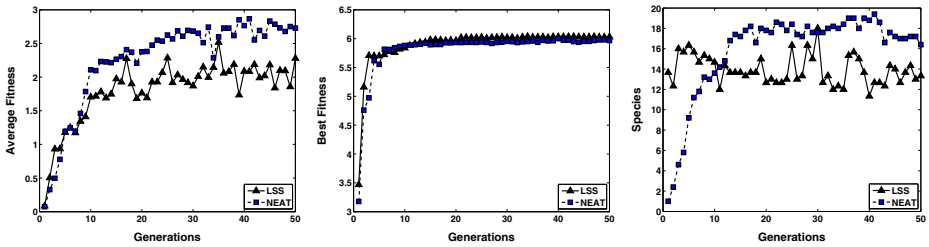


Fig. 12 Each row shows a different set of *singular behaviors* obtained with the LSS measure for the 4-circle world. Notice the diversity of navigation strategies that are possible within this environment



(a) Average fitness (b) Best fitness (c) No.of species

Fig. 13 Performance plots that compare behavior-based speciation with the LSS measure and the NEAT method for the homing navigation problem with the Khepera robot. All plots are averages over the total number of runs

presented in Table 1, except that the total number of runs was set to five and the behavior threshold is $h = 5.3$.

5.3.1 Evolution Statistics

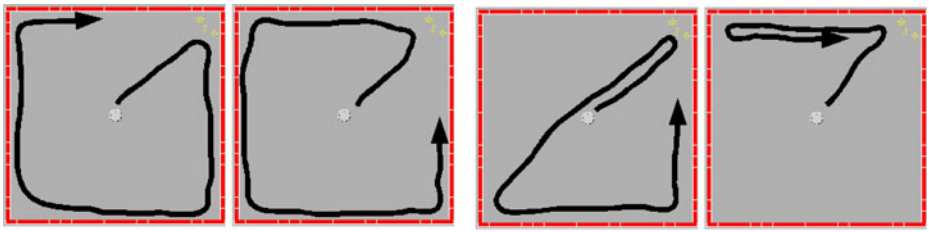
Figure 13a–c present comparative plots between LSS and NEAT. All graphs are plotted relative to the number of generations and represent the average over all the runs. The plots show comparisons based on the number of species that each method generates, the best individual fitness, and the average population fitness. In terms of fitness, both methods again produce similar results, in both cases we can see that the evolutionary process quickly converges. It appears that solving this problem is not a difficult task for the ER system. However, with respect to the total number of species we can see that the NEAT methods generates more species than does LSS. For LSS, the number of species oscillates around 15, consistent with the previous experiments. Therefore, even do both methods solve this multimodal problem quite easily, the speciation methods are producing different results.

5.3.2 Behavior-based Speciation

There are some noticeable differences between the speciation that each method produces, see Table 6. On the one hand, both LSS and NEAT generate a similar amount of singular behaviors, with the former generating two in every run, and the latter generating two or three. On the other hand, the number of species behaviors is quite different, with NEAT generating more than five times as much as LSS. This discrepancy is evident in the $\mathcal{BSR}$ of each. NEAT achieves a very small $\mathcal{BSR}$ value, this suggests that many species converge towards the same behavior or are redundant. Conversely, the LSS measure consistently achieves an ideal $\mathcal{BSR} = 1$,

Table 6 Statistical comparison of LSS and NEAT for homing navigation with the Khepera

Measure	$\lvert\mathcal{B}\rvert$		$\lvert\mathcal{I}\rvert$		$\mathcal{BSR}$
	μ	σ	μ	σ	
LSS	2	0	2	0	1
NEAT	10.4	0.547	2.6	0.547	0.25



(a) (b)

Fig. 14 Examples of the singular behaviors found with the LSS measure for the homing navigation problem. Each figure shows two singular behaviors found in different runs of the algorithm

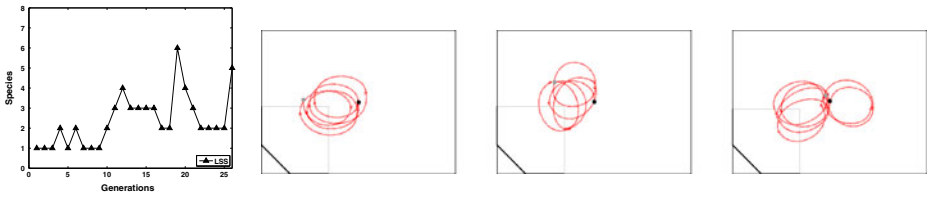
thus LSS always finds unique behaviors within each species it generates. Figure 14 shows the singular behaviors obtained with LSS in two different runs.

5.4 Homing Navigation: Pioneer P2-AT

In this final experiment, we are interested in testing the generality and scalability of our proposal. Therefore, we apply behavior-based speciation on the homing navigation problem using a different robot, the Pioneer P2-AT; see Section 4.4.2. In this case, we only test our proposal with the LSS measure and evolve the neurocontrollers using the Saphira simulator and Colbert programming language provided by ActivMedia Robotics. Finally, the scalability of the evolved behaviors was tested by deploying them onto a real robot.

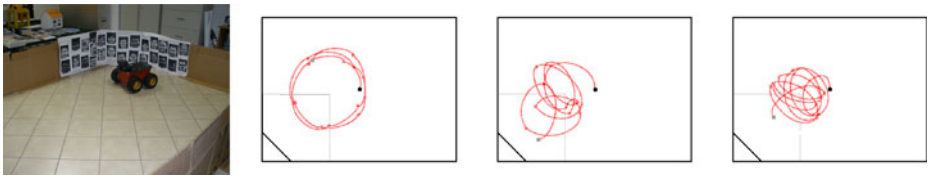
5.4.1 Simulation

The algorithm was executed using similar parameter values to those in Table 1. However, the Pioneer simulator only runs in real-time, and the evolutionary process can last for as long as one week. Therefore, the size of the population was set to 40 individuals and the total number of generations was reduced to 25. Figure 15a shows a plot of how the total number of species varied for a single run. From this example, evolution produced three different robot behaviors that solve the homing navigation problem, Fig. 15b–d shows each behavior.



(a) Species (b) Behavior 1 (c) Behavior 2 (d) Behavior 3

Fig. 15 **a** Species formation for homing navigation and the Pioneer robot. **b–d** The three singular behaviors generated for homing navigation with the simulated Pioneer P2-AT; *dark point* represents the initial position



(a) Real Environment (b) Behavior 1 (c) Behavior 2 (d) Behavior 3

Fig. 16 The three singular behaviors generated for the homing navigation problem transfered onto the real Pioneer P2-AT

5.4.2 Real-World Deployment

After evolving the robot behaviors offline, we have transfered the three singular behaviors onto a real Pioneer P2-AT robot. Figure 16a shows the real-world environment used to test the evolved controllers, and Fig. 16b–d show the path generated by each within this environment. Notice that the simulated and the paths are quite similar, except for behavior 3, and in all cases we observe that the robot performs a simple movement that periodically recharges the simulated battery.

6 Summary and Concluding Remarks

In evolutionary robotics, the main goal is to use artificial evolution to automatically generate robot behaviors that perform a specific task. However, because an implicit task specification is employed, it is difficult to know the structure of the solution space beforehand. Therefore, if several different behaviors were obtained, these would provide a better understanding of how the problem could be solved. This paper describes a behavior-based speciation method that encourages several navigation strategies to evolve concurrently within a single population. Behaviors are compared using *behavior signatures* which represent a path followed by the robot within the environment. Signatures are expressed as character strings and several similarity measures were tested with the speciation method. The proposed behavior-based speciation was compared with the topology-based speciation used by the NEAT method and with a canonical genetic algorithm.

Through behavior-based speciation the evolutionary process produced several navigation strategies, each exhibiting a different structure. The speciation process coherently divided the population based on behavioral characteristics and forced species to converge towards different types of behaviors. The algorithm found a diverse set of unique behaviors that achieve the same task, called *singular behaviors*.

We have also confirmed that the diversity of behaviors did not depend upon large topological differences between the neurocontrollers. In fact, in some cases different behaviors were obtained from ANNs that share a similar topology. Furthermore, results indicate the occurrence of an unexpected phenomenon. Behaviors generated with the behavior-based speciation were more robust when placed with an unknown environment. The speciation process did not allow evolution to over-fit the population to the training environment.

Additionally, the generality of the speciation method was experimentally confirmed by applying it to different problems, using several training environments, and

two different robots. Moreover, good scalability was shown by deploying some of the evolved behaviors onto a real robot.

References

1. Coello, C., Veldhuizen, D.V., Lamont, G.: Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic Publishers, New York, New York (2002)
2. Darwen, P.J., Yao, X.: Speciation as automatic categorical modularization. IEEE Trans. Evol. Comput. **1**(2), 101–108 (1997)
3. DeJong, K.A.: Evolutionary Computation: A Unified Approach. MIT Press, Cambridge, MA, USA (2002)
4. Dorigo, M., Stützle, T.: Ant Colony Optimization. Bradford Company, Scituate, MA, USA (2004)
5. Dunn, E., Olague, G., Lutton, E.: Parisian camera placement for vision metrology. Pattern Recogn. Lett. **27**(11), 1209–1219 (2006)
6. Eberhart, R.C., Shi, Y., Kennedy, J.: Swarm Intelligence, 1st edn. The Morgan Kaufmann Series in Evolutionary Computation, Morgan Kaufmann (2001)
7. Floreano, D., Mattiussi, C.: Bio-inspired Artificial Intelligence: Theories, Methods and Technologies. MIT Press, Cambridge, MA (2008)
8. Floreano, D., Sanderson, F.M.A.C.: Evolution of homing navigation in a real mobile robot. IEEE Trans. Syst. Man Cybern. Part B **26**(3), 396–407 (1996)
9. Goldberg, D.E., Richardson, J.: Genetic algorithms with sharing for multimodal function optimization. In: Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their Application, pp. 41–49. Lawrence Erlbaum Associates, Inc., Mahwah, NJ, USA (1987)
10. Gomez, F.J., Miikkulainen, R.: Solving non-Markovian control tasks with neuro-evolution. In: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI 99, pp. 1356–1361. Morgan Kaufmann, San Mateo, CA (1999)
11. Hocaoğlu, C., Sanderson, A.C.: Planning multiple paths with evolutionary speciation. IEEE Trans. Evol. Comput. **5**(3), 169–191 (2001)
12. Landrin-Schweitzer, Y., Collet, P., Lutton, E., Prost, T.: Introducing lateral thinking in search engines with interactive evolutionary algorithms. In: SAC '03: Proceedings of the 2003 ACM Symposium on Applied Computing, pp. 214–219. ACM Press, New York, NY, USA (2003)
13. Mahfoud, S.W.: Niching methods for genetic algorithms. Ph.D. thesis, University of Illinois at Urbana-Champaign, Champaign, IL, USA (1995)
14. Martin, P., Bateson, P.: Measuring Behaviour: An Introductory Guide, 3rd edn. Cambridge University Press, Cambridge, UK (2007)
15. Mattiussi, C., Waibel, M., Floreano, D.: Measures of diversity for populations and distances between individuals with highly reorganizable genomes. Evol. Comput. **12**(4), 495–515 (2004)
16. Michel, O.: Khepera Simulator v. 2 User Manual. University of Nice-Sophia, Antipolis (1996)
17. Miglino, O., Lund, H.H., Nolfi, S.: Evolving mobile robots in simulated and real environments. Artif. Life **2**(4), 417–434 (1995)
18. 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 Kaufman, San Francisco, California (1989)
19. Moriarty, D.E., Mikkulainen, R.: Efficient reinforcement learning through symbiotic evolution. Mach. Learn. **22**(1–3), 11–32 (1996)
20. Nguyen, Q.H., Ong, Y.S., Lim, M.H.: A probabilistic memetic framework. IEEE Trans. Evol. Comput. **13**, 604–623 (2009)
21. Nitschke, G., Schut, M.: Designing multi-rover emergent specialization. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2008). ACM Press (2008)
22. Nolfi, S., Floreano, D.: Evolutionary Robotics: the Biology, Intelligence, and Technology of Self-Organizing Machines. Bradford Book (2004)
23. Ong, Y.S., Lim, M.H., Chen, X.: Research frontier: memetic computation-past, present & future. IEEE Comput. Intell. Mag. **5**, 24–31 (2010)
24. Pollack, J.B., Blair, A.D.: Co-evolution in the successful learning of backgammon strategy. Mach. Learn. **32**(1), 225–240 (1998)

25. Potter, M.: The design and analysis of a computational model of cooperative coevolution. Ph.D. thesis, George Mason University, Fairfax, VA, USA (1997)
26. Price, K., Storn, R.M., Lampinen, J.A.: Differential Evolution: a Practical Approach to Global Optimization (Natural Computing Series). Springer-Verlag New York, Inc., Secaucus, NJ, USA (2005)
27. Rosca, J.: Hierarchical learning with procedural abstraction mechanisms. Ph.D. thesis, Rochester, NY, USA (1997)
28. Savage, T.: Measurement and the explanation of adaptive and novel behaviors in real and artificial creatures. Cogn. Syst. Res. **5**(1), 3–39 (2004)
29. Stanley, K.O., Miikkulainen, R.: Evolving neural networks through augmenting topologies. Evol. Comput. **10**(2), 99–127 (2002)
30. Trujillo, L., Olague, G., Lutton, E., de Vega, F.F.: Discovering several robot behaviors through speciation. In: Giacobini, M., et al. (eds.) EvoWorkshops: the 4th European Workshop on Bio-inspired Heuristics for Design Automation (EvoHOT'07). Lecture Notes in Computer Science, vol. 4974, pp. 164–174, 26–28 March, Napoli, Italy, Springer (best paper award) (2008)
31. Viola, P.A., Jones, M.J.: Rapid object detection using a boosted cascade of simple features. In: Proceeding from the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), pp. 511–518, 8–14 December, Kauai, HI, USA. IEEE Computer Society (2001)
32. Yujian, L., Bo, L.: A normalized levenshtein distance metric. IEEE Trans. Pattern Anal. Mach. Intell. **29**(6), 1091–1095 (2007)