Behavior-based Speciation in Classification with NeuroEvolution

Evgenia Papavasileiou
(1) Department of Electronics and Informatics (ETRO)
Vrije Universiteit Brussel (VUB)
Pleinlaan 2, B-1050, Brussels, Belgium
(2) imec
Kapeldreef 75, B-3001 Leuven, Belgium
epapavas@etrovub.be

Jan Cornelis
Department of Electronics and Informatics (ETRO)
Vrije Universiteit Brussel (VUB)
Pleinlaan 2, B-1050, Brussels, Belgium
jpcornel@etrovub.be

Bart Jansen
(1) Department of Electronics and Informatics (ETRO)
Vrije Universiteit Brussel (VUB)
Pleinlaan 2, B-1050, Brussels, Belgium
(2) imec
Kapeldreef 75, B-3001 Leuven, Belgium
bjansen@etrovub.be

***Abstract*—We propose a neuroevolutionary speciation mechanism that is applied on NeuroEvolution of Augmenting Topologies (NEAT) that solely evolves neural networks' topology and weights and its extension HA-NEAT that also evolves activation functions. The new speciation mechanism is defined based on the behavior of the individuals rather than their topological similarity. Focusing on classification tasks we build artificial datasets of high complexity. Performance is described by (i) median classification accuracy, (ii) computational efficiency (number of generations) and (iii) network complexity (number of nodes and connections). The performance metrics are compared using Kruskal-Wallis hypothesis tests with Bonferroni correction. It is found that the proposed behavioral speciation mechanism outperforms the original speciation solving problems that were not solvable before or improving the accuracy and reducing the network complexity.**

***Index Terms*—NeuroEvolution, speciation, behavior, classification**

I. Introduction

NeuroEvolution (NE) is a learning method that uses Evolutionary Algorithms (EAs) to optimize parameters of Artificial Neural Networks (ANNs) [1]. NeuroEvolution of Augmenting Topologies (NEAT) [2] is a neuroevolutionary method for optimizing both the connection weights and the topology of ANNs. NEAT introduces three main innovations (section II) that solve significant problems in the field of NE such as the competing conventions problem and facilitates the crossover between individuals of different length. Moreover, NEAT evolves networks by adding new structure when necessary and protects new individuals by organizing them in species. Heterogeneous Activation NEAT (HA-NEAT) [3], that evolves ANNs with mixed activation functions, is one of the many methods [3]–[5] proposed to extend NEAT's functionality.

Speciation or niching is a mechanism of protecting innovation inspired by nature. Individuals are assigned into species based on a similarity metric. In NEAT and HA-NEAT this metric assesses how similar two topologies are, based on how matching or disjoint the connection genes of two individuals are. However, especially in HA-NEAT that evolves ANNs with nodes of different activation functions, the definition of a similarity metric based only on the connection genes is not sufficient to capture the differences and similarities of the evolved heterogeneous topologies. Moreover, the definition of the topological distance metric is based on the assumption that similar genotypes correspond to similar phenotypes. However, a mutation operation in the genome that may affect only a little this distance metric can have a huge impact on the network's output, i.e. on the network's functionality. For example, removing a connection gene only slightly changes the genotype, but the impact on the phenotype can be huge as a network's output can change completely, e.g., a whole part of a network could become disconnected from the output. In this paper we propose a speciation mechanism for assigning individuals into species based on their 'behavior'. Since we are focusing on classification tasks, we define as a network's behavior the predicted output for a given input. Speciation is now performed in the behavioral space rather than the topological space and this type of speciation is called 'behavior-based speciation'.

The proposed speciation scheme is tested on both NEAT and HA-NEAT and it is compared with the original speciation scheme of both NEAT and HA-NEAT. The four resulting algorithms are evaluated on artificial datasets of high complexity in two aspects; their classification ability in terms of accuracy and their efficiency, measured by the number of generations required until the stopping criterion is met and the size of the evolved networks (number of hidden nodes and connections).

II. Methods

A. NEAT

NEAT [2] is a successful neuroevolutionary method that allows the simultaneous optimization of the connectivity and the topology of ANNs. NEAT uses direct encoding of the networks with a set of node genes and a set of connection genes that undergo crossover and mutation operations. Two types of mutations are allowed structural and weight mutations. Structural mutations enable the addition of new nodes

This work was funded by a PhD grant awarded to Evgenia Papavasileiou from the Research Foundation Flanders (FWO), scholarship numbers 1S34016N, 1S34018N.

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

and connections, while weight mutations perturb the weights of existing connections.

NEAT introduces three main innovations. First of all, it enables the alignment of matching genes during crossover by historical markings, more precisely by assigning an innovation number that is incremented each time a new gene is added to the genome thus functioning as a chronological parameter. The innovation number is then inherited by the offspring and maintained throughout evolution. Furthermore, NEAT performs speciation by dividing the population into groups based on their topological similarity. In this way, a new individual can compete within its own niche and it is given time to be optimized, instead of having to compete with other already evolved networks in the population. Finally, NEAT starts the evolution with a population of minimal structures, i.e. a population of networks with only fully connected input and output layers. Gradually, it evolves more complex networks by introducing new nodes and connections that survive only if they increase the performance, measured by a fitness function, which means that NEAT tends to discover networks without unnecessary structure.

B. HA-NEAT

HA-NEAT [3] is an extension of the NEAT algorithm that allows the evolution of ANNs with mixed activation functions. Hagg et al. [3] showed, that in order to approximate a function, more nodes of the same activation function are required compared to when using nodes of different activation functions. As a result, when heterogeneous networks are evolved, they are supposed to be smaller, having fewer parameters to optimize so being less prone to overfitting. In HA-NEAT, NEAT's 'add-node' mutation operator is altered to introduce a new node in the genome with a random activation function from a predefined list of possible functions. This set can contain the step function, the Rectifier Linear Unit function (RELU), the sigmoid and the Gaussian functions. Moreover, a new mutation operator is introduced, called 'mutate activation function'. This operator selects a random node of the genome and changes its activation function by choosing a new function from the previously described predefined set of possible functions.

III. Speciation

Speciation in nature is the darwinian phenomenon of an evolving population consisting of different subgroups whose individuals share a characteristic that is different from the one of individuals of other subgroups [6]. Speciation or niching in Genetic Algorithms and NE is inspired by speciation in nature. It is a mechanism for protecting innovation, as new structures can be isolated in their own species and get time to optimize their structure before competing with the whole population [2].

A. Speciation in NEAT and HA-NEAT

In NEAT and HA-NEAT individuals are assigned into species based on a compatibility/similarity distance between the individual and a representative of each species. If the distance is smaller than a compatibility threshold, the individual belongs to that species, otherwise a new species is created. The definition of this distance metric is based on the assumption that the more evolutionary history two genomes share, the more compatible they are [2]. This is evaluated with the help of the historical markings that facilitate the aligning of the connection genes of two individuals. The non matching genes between the two individuals that are located in the middle of the genomes are called disjoint genes, while the non matching genes in the end of the genomes are called excess genes. The compatibility distance, defined in equation 1, is calculated based on the number of excess and disjoint genes and the average weight difference between matching genes.

$$\delta = c_1 \cdot E + c_2 \cdot D + c_3 \cdot \overline{W} \tag{1}$$

with E, the number of excess genes, D the number of disjoint genes, $\overline{W}$ the average weight difference between matching genes and c_1, c_2, c_3 coefficients that determine the importance of each factor.

B. Proposed Behavioral Speciation

However, in the case of HA-NEAT that evolves networks whose nodes have different activation functions, a distance metric evaluated only on matching/mismatching connection genes is not sufficient to capture the similarity of two networks. In addition, a speciation metric evaluating the similarity of genotypes is based on the assumption that similarity at the level of genotypes corresponds to similarity at the level of phenotypes. Nevertheless, a mutation operation that can have a small impact on a metric evaluated on the distance between genotypes can have a big influence in the resulting phenotype and hence in the functionality of the network. Therefore, evaluating the similarity of networks at the level of genotypes cannot capture how similar the networks function.

In this paper we propose a speciation mechanism based on the functional behavior of the ANNs rather than their genomes' similarities. Since we are focusing on classification tasks we define as a network's behavior the predicted output for a given input. In our approach, the compatibility distance/similarity metric between two individuals, defined in equation 2, is given by the sum of the absolute difference of their outputs.

$$\forall i, j \in population$$
$$\delta_b(i,j) = \sum_{\forall samples} |output(i) - output(j)| \tag{2}$$

with $\delta_b(i,j)$ the behavioral distance between two individuals i and j and $output(i)$, $output(j)$ their corresponding output vectors.

The individuals are then assigned into species by k-means clustering using a squared Euclidean distance metric and k-means++ algorithm [7] to initialize the clusters' centers.

In this paper we apply behavioral speciation both on HA-NEAT and NEAT and we call these algorithms Behavioral-based Speciation-HA-NEAT (BS-HA-NEAT) and Behavioral-

based Speciation-NEAT (BS-NEAT) respectively. In [8], we experimentally proved that using the Gaussian activation function in the output layer of FD-NEAT [9] (another extension of NEAT), rather than the traditionally used sigmoid functions, results in better performance (accuracy, number of generations and size of the evolved networks). We follow this approach here so that (BS)-NEAT and (BS)-HA-NEAT evolve ANNs with Gaussian activation functions in the output layer.

The concept of 'behavioral speciation' has been encountered in NEAT-based [6], [10] and HyperNEAT-based [11] methods before, however behavior was defined differently [6], [10] or it was employed in a different algorithmic context [11].

IV. Experimental Setup

A. Datasets

The first step towards evaluating the performance of a new method is to test it on artificial data whose expected behaviour is known, as they allow the true assessment of the method's performance compared to real-world scenarios whose true solution is often unknown [12]. XOR is a well-known, non-linearly separable problem and one of the first datasets researchers would consider to verify the success of their approach [2], [9], [13]. Since original XOR of 4 samples and 2 inputs can be solved by an ANN with one hidden layer, we build much more complex datasets with many irrelevant inputs, named 2 out of k datasets (referred to as $2/k$), where k is the total number of features, following an approach described in [8]. The 2 inputs are assigned to the relevant features and the remaining $k-2$ inputs are assigned to irrelevant binaries. In this paper we choose the total number of features to be equal to $k = 100$. To include these irrelevant features we increase the number of samples to reduce the probability that an underlying correlation exists between the randomly generated data and the output. In this way we create datasets of increasing complexity with 700, 500, 300, and 100 samples of 100 inputs to which we refer as $2/100 - N_s$, ($N_s \in \{700, 500, 300, 100\}$). The smaller the proportion between dataset's size and number of features is, the more difficult the problem becomes.

Simpler versions of these datasets have been used as benchmark problems in feature selection tasks [8] as the datasets are constructed such that each of the individual attributes is equally informative for predicting the output. The datasets constructed in this paper are even more complex and more difficult to be learnt as we have both significantly increased the number of irrelevant features and decreased the number of samples.

Spiral plots constitute another family of artificial datasets, versions of which have been used to benchmark FD-NEAT [9]. We construct datasets of 1000 samples and two inputs that are characterized by a highly-non-linear decision boundary, as the one depicted in Figure 1.

We repeat the dataset construction five times and divide the resulting $2/100 - N_s$ XOR and spiral datasets into training and test sets by 10 fold cross validation.

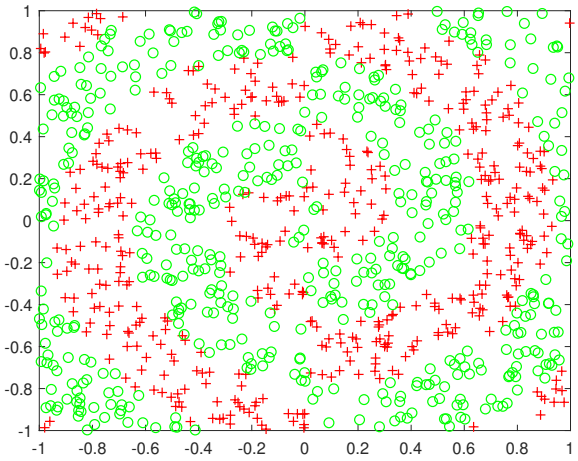


Fig. 1: Example of a spiral plots dataset

Performing 10 fold cross validation on the $2/100-100$ XOR results in training sets of 90 samples and 100 features which are very difficult to learn, as there are not enough samples to capture the underlying relationship between features and classes.

B. Genetic Algorithm Settings

The fitness function used in the (BS)-NEAT and (BS)-HA-NEAT algorithms is defined by the error between the output of the ANN and the real label of the training sample. In this way the fitness function is defined as in equation 3.

$$fitness = \frac{1}{error} = \frac{1}{1 - accuracy} = \frac{1}{1 - \frac{N_{cc}}{N_{tr}}}. \quad (3)$$

with N_{cc} the number of samples that are classified correctly and N_{tr} the number of training samples.

The stopping criterion of evolution is defined when the fitness value becomes greater than $1/0.01$ i.e. when the training error is smaller than 0.01 or when the maximum number of allowed generations is reached.

NEAT and HA-NEAT with the two different speciation mechanisms run with the same parameter configurations (Table I) for all the $2/100 - N_s$ XOR and spiral plots datasets.

C. Implementation

The proposed speciation scheme as well as HA-NEAT and NEAT are implemented in MATLAB based on a MATLAB implementation of HA-NEAT [14].

D. Evaluation Measures

The $2/100 - N_s$ XOR datasets and the spiral datasets are divided into training and test sets by 10 fold cross validation and the experiments are repeated 5 times, resulting in 50 runs of each of the four algorithms (NEAT, BS-NEAT, HA-NEAT and BS-HA-NEAT). The performance of the algorithms is evaluated on two aspects; their classification ability and their efficiency. The classification ability is evaluated using

TABLE I: Parameter Setting of (BS)-NEAT and (BS)-HA-NEAT for the $2/100 - N_s$ XOR and Spiral Plots Tasks

Parameter	Value	Meaning
Population Size	350	The number of individuals in the population
Max generations	1000	The maximum number of generations allowed
P_r(crossover)	0.8	The probability of performing crossover
P_r(add node)	0.4	The probability of adding a new node
P_r(add connection)	0.05	The probability of adding a new connection
P_r(mutate weight)	0.1	The probability of changing the weight of a connection
P_r(mutate activation function)	0.2	The probability of changing the activation function of a node (in HA-NEAT)
P_r(enable connection)	0.005	The probability of re-enabling a disabled connection
P_r(disable connection)	0.005	The probability of disabling an enabled connection
Coefficient 1	1	Compatibility coefficient of excess genes
Coefficient 2	1	Compatibility coefficient of disjoint genes
Coefficient 3	0.4	Compatibility coefficient of average weight difference

the *accuracy* on the test set, while their efficiency is based on the number of *generations* required until the stopping criterion is met and the size of the evolved topologies, in terms of the number of evolved *nodes* and *connections*.

For each of these metrics we calculate and present the median values and the inter quartile range (IQR) over the 50 runs. We chose to calculate the median values instead of the average because the presence of outliers can influence the interpretation of the results.

E. Analysis of the Results

We perform statistical tests to compare the performance of NEAT and HA-NEAT with the two speciation mechanisms using Kruskal-Wallis hypothesis tests with Bonferonni correction ($p < 0.01$). The hypothesis tests are applied on the different performance metrics mentioned in the previous section; classification accuracy, number of generations, number of nodes and number of connections.

V. Results

Tables II-III present the median values of the investigated performance metrics and the corresponding IQR calculated over 50 runs on the different $2/100 - N_s$ XOR problems with sample size $N_s \in \{700, 500, 300, 100\}$. For details of the distribution of the results we present Figures 2 and 3 that show the performance of all the algorithms on all the $2/100 - N_s$ XOR datasets. Each of the plots presents the values of a performance metric (y axis) for the HA-NEAT (Figure 2) and NEAT (Figure 3) for the two different speciation mechanisms tested across the different datasets (x axis). Moreover, the p values of the hypothesis tests are presented in Tables V, VI, VII and VIII for each of the $2/100 - N_s$ XOR datasets. Each row and column includes the methods whose performance metrics are tested, i.e. NEAT, BS-NEAT, HA-NEAT and BS-HA-NEAT. For illustrating purposes, we present the results that are statistically different on a grey background. The results concerning the spiral plots problems can be found in Tables IV and IX.

TABLE II: Performance Analysis of HA-NEAT with Original Speciation and HA-NEAT with Behavioral Speciation (BS-HA-NEAT) on the $2/100 - N_s$ XOR Datasets

Metric	——————-Dataset's difficulty increases to the right ⟶ $2/100 - N_s$ XOR datasets with N_s= 700	500	300	beyond feasible 100
HA-NEAT Original Speciation				
Accuracy	0.53 (0.13)	0.52 (0.08)	0.5 (0.07)	0.5 (0.20)
Generations	1000 (0)	1000 (0)	1000 (0)	1000 (0)
Hidden Nodes	77.5 (58)	42.5 (70)	52 (78)	35 (54)
Connections	234 (168)	136.5 (122)	170.5 (136)	123 (72)
HA-NEAT Behavioral Speciation (BS-HA-NEAT)				
Accuracy	0.99 (0.03)	0.98 (0.10)	0.88 (0.47)	0.5 (0.20)
Generations	643 (301)	726 (441)	1000 (317)	1000 (0)
Hidden Nodes	46 (12)	49.5 (14)	47.5 (16)	56 (23)
Connections	125.5 (32)	136.5 (48)	136.5 (36)	163 (50)

TABLE III: Performance Analysis of NEAT with Original Speciation and NEAT with Behavioral Speciation (BS-NEAT) on the $2/100 - N_s$ XOR Datasets

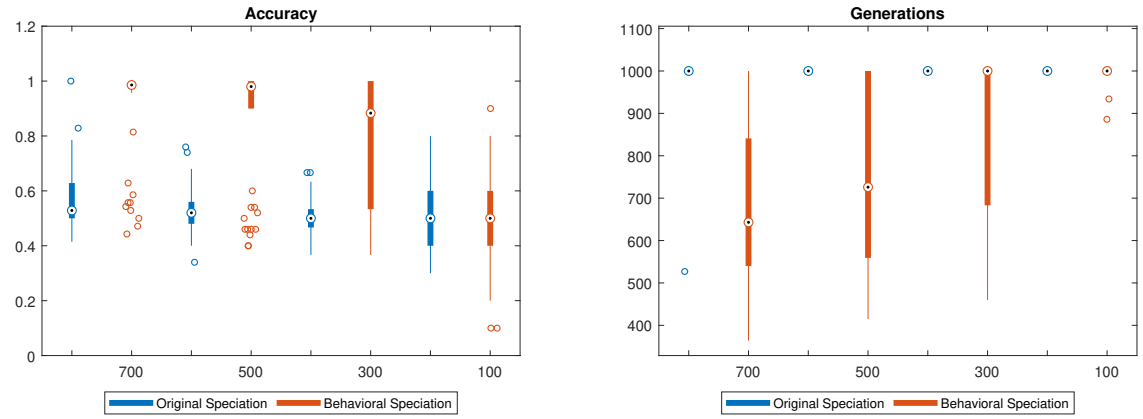
Metric	———-Dataset's difficulty increases to the right ⟶ $2/100 - N_s$ XOR datasets with N_s= 700	500	300	beyond feasible 100
NEAT Original Speciation				
Accuracy	0.50 (0.07)	0.52 (0.06)	0.5 (0.07)	0.5 (0.07)
Generations	1000 (0)	1000 (0)	1000 (0)	1000 (0)
Hidden Nodes	49.5 (94)	40 (69)	47.5 (71)	49.5 (94)
Connections	159.5 (175)	147.5 (98)	153 (110)	159.5 (175)
NEAT Behavioral Speciation (BS-NEAT)				
Accuracy	0.98 (0.29)	0.81 (0.48)	0.60 (0.47)	0.5 (0.2)
Generations	842.5 (338)	1000 (211)	1000 (114)	1000 (0)
Hidden Nodes	50 (21)	60 (23)	49.5 (19)	50.5 (21)
Connections	137.5 (44)	161.5 (48)	145 (44)	151 (44)

TABLE IV: Performance Comparison of NEAT and HA-NEAT with Original and Behavioral Speciation on the Spiral Plots Datasets

Metric	NEAT		HA-NEAT	
	Original	**Behavioral**	**Original**	**Behavioral**
Accuracy	0.67 (0.08)	0.77 (0.1)	0.66 (0.08)	0.7 (0.08)
Generations	1000 (0)	1000 (0)	1000 (0)	1000 (0)
Hidden Nodes	100 (43)	64.5 (25)	76 (41)	55.5 (22)
Connections	237.5 (115)	151.5 (49)	176.5 (104)	132.5 (66)

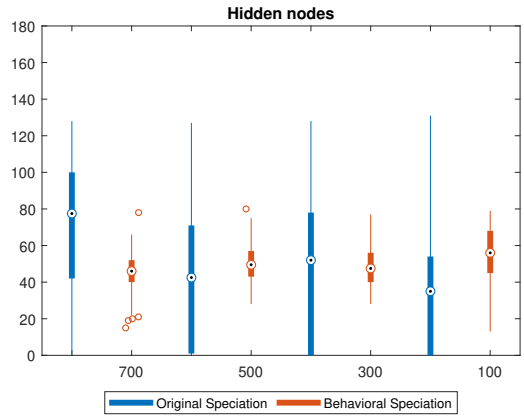
We observe that both NEAT and HA-NEAT are unable to solve any of the $2/100 - N_s$ XOR problems ($N_s \in \{700, 500, 300\}$) reaching an accuracy of median value around 0.5, which means that the algorithms predict the output randomly. On the other hand, the proposed speciation mechanism applied on both HA-NEAT and NEAT results in significantly different performance ($p < 0.01$) than the original speciation. BS-NEAT and BS-HA-NEAT solve the three problems with significantly higher accuracy in significantly less generations than NEAT and HA-NEAT with original speciation respectively. From the results obtained for the $2/100 - 100$ XOR problem we can also verify our initial hypothesis that the constructed problem is no longer feasible to be solved, as all the methods reach a median accuracy of 0.5.

Concerning the spiral plots problem, the accuracy obtained by BS-HA-NEAT and BS-NEAT is significantly higher than

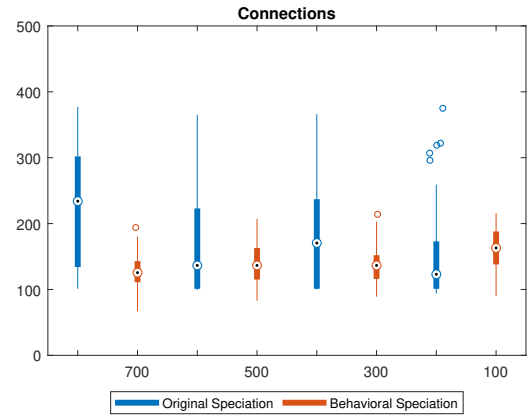


(a) Accuracy

(b) Generations



(c) Number of Nodes



(d) Number of Connections

Fig. 2: Accuracy on the test set, Generations, Number of Evolved Nodes and Connections for HA-NEAT and BS-HA-NEAT on the different $2/100 - N_s$ XOR datasets.

HA-NEAT's ($p = 0.04$) and NEAT's ($p = 0$) respectively. However, none of the algorithms solves the problem within the predefined number of allowed generations. This might be the result of using the same parameter settings (Table I) for both the $2/100-N_s$ XOR and spiral plots tasks without optimizing the values for each problem separately.

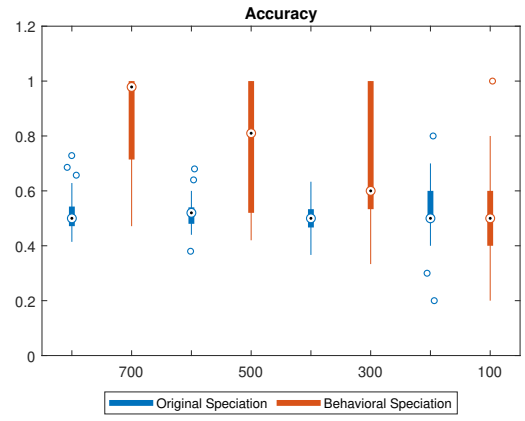
Moreover, the networks evolved by BS-HA-NEAT and BS-NEAT are smaller than the networks evolved by HA-NEAT and NEAT. This difference is statistically different in the case of spiral plots but not in the case of the $2/100 - N_s$ XOR problems, except for some cases in the $2/100 - 700$ XOR problem as can be observed in Table V.

In addition, on the $2/100 - N_s$ XOR problems ($N_s \in \{700, 500, 300\}$), we observe that the accuracy of both BS-NEAT and BS-HA-NEAT decreases as the number of samples N_s decreases, while the number of generations required to find the solution increases. This is an expected behavior, because the smaller the number of samples a dataset has, the more difficult the dataset becomes and more generations are required to find the solution.

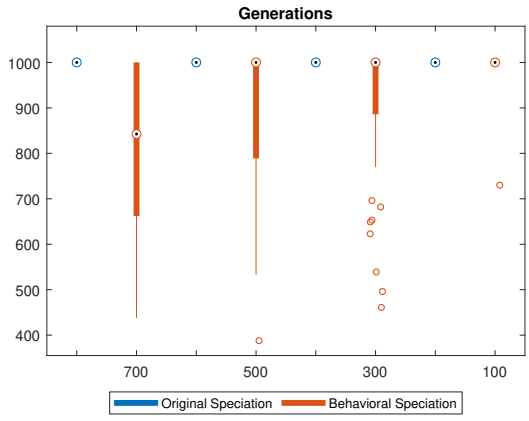
Comparing the performance of BS-HA-NEAT with BS-NEAT there is no clear conclusion as far as the accuracy is concerned. On the $2/100 - N_s$ XOR problems ($N_s \in \{700, 500, 300\}$) it looks as the former performs better than the latter, but no significant difference exists, whereas on the spiral plots BS-NEAT performs significantly better than BS-HA-NEAT. Finally, regarding their efficiency, in some cases BS-HA-NEAT is more efficient than BS-NEAT as it requires significantly less generations (e.g. for the $2/100 - 700$ XOR ($p = 0.012$) and the $2/100 - 500$ XOR ($p < 0.01$)), but no significant difference exists regarding the size of the networks.

VI. Conclusion

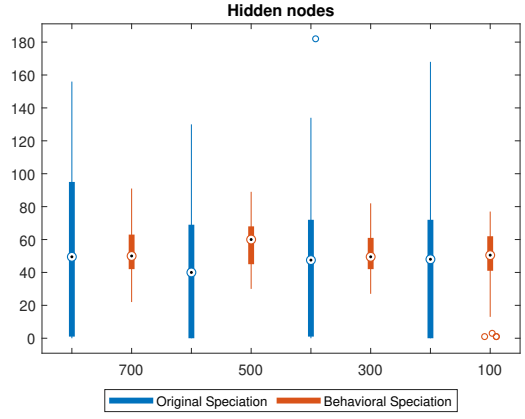
In this paper we proposed a neuroevolutionary speciation mechanism based on individuals' behavior. We tested this behavioral speciation mechanism both for NEAT and HA-NEAT on highly-non-linear classification tasks and we showed



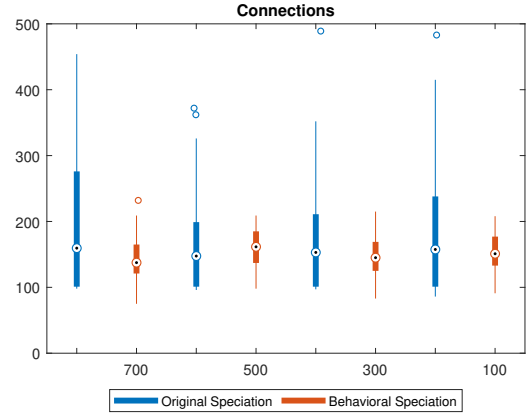
(a) Accuracy



(b) Generations



(c) Number of Nodes



(d) Number of Connections

Fig. 3: Accuracy on the test set, Generations, Number of Evolved Nodes and Connections for NEAT and BS-NEAT on the different $2/100 - N_s$ XOR datasets.

TABLE V: P value of Kruskal Wallis hypothesis tests with Bonferonni Correction for the 2/100-700 XOR Dataset

	Metric	NEAT	BS-NEAT	HA-NEAT	BS-HA-NEAT
NEAT	accuracy	-	0	0.42	0
	generations	-	0	1	0
	hidden nodes	-	1	0.14	0.72
	connections	-	0.41	0.19	0
BS-NEAT	accuracy	0	-	0	1
	generations	0	-	0	0.012
	hidden nodes	1	-	0.10	0.94
	connections	0.41	-	0	0.35
HA-NEAT	accuracy	0.42	0	-	0
	generations	1	0	-	0
	hidden nodes	0.14	0.10	-	0
	connections	0.19	0	-	0
BS-HA-NEAT	accuracy	0	1	0	-
	generations	0	0.012	0	-
	hidden nodes	0.72	0.94	0	-
	connections	0	0.35	0	-

TABLE VI: P value of Kruskal Wallis hypothesis tests with Bonferonni Correction for the 2/100-500 XOR Dataset

	Metric	NEAT	BS-NEAT	HA-NEAT	BS-HA-NEAT
NEAT	accuracy	-	0	1	0
	generations	-	0	1	0
	hidden nodes	-	0.03	1	1
	connections	-	1	1	1
BS-NEAT	accuracy	0	-	0	0.48
	generations	0	-	0	0
	hidden nodes	0.03	-	0.10	0.27
	connections	1	-	1	0.08
HA-NEAT	accuracy	1	0	-	0
	generations	1	0	-	0
	hidden nodes	1	0.10	-	1
	connections	1	1	-	0.88
BS-HA-NEAT	accuracy	0	0.48	0	-
	generations	0	0	0	-
	hidden nodes	1	0.27	1	-
	connections	1	0.08	0.88	-

TABLE VII: P value of Kruskal Wallis hypothesis tests with Bonferonni Correction for the 2/100-300 XOR Dataset

	Metric	NEAT	BS-NEAT	HA-NEAT	BS-HA-NEAT
NEAT	accuracy	-	0	1	0
	generations	-	0	1	0
	hidden nodes	-	1	1	1
	connections	-	1	1	1
BS-NEAT	accuracy	0	-	0	0.74
	generations	0	-	0	0.21
	hidden nodes	1	-	1	1
	connections	1	-	1	1
HA-NEAT	accuracy	1	0	-	0
	generations	1	0	-	0
	hidden nodes	1	1	-	1
	connections	1	1	-	0.18
BS-HA-NEAT	accuracy	0	0.74	0	-
	generations	0	0.21	0	-
	hidden nodes	1	1	1	-
	connections	1	1	0.18	-

TABLE VIII: P value of Kruskal Wallis hypothesis tests with Bonferonni Correction for the 2/100-100 XOR Dataset

	Metric	NEAT	BS-NEAT	HA- NEAT	BS-HA-NEAT
NEAT	accuracy	-	1	1	1
	generations	-	1	1	1
	hidden nodes	-	1	0.20	0.26
	connections	-	1	0.33	1
BS-NEAT	accuracy	1	-	1	1
	generations	1	-	0.61	1
	hidden nodes	1	-	0.06	0.71
	connections	1	-	0.63	1
HA-NEAT	accuracy	1	1	-	1
	generations	1	0.61	-	1
	hidden nodes	0.20	0.06	-	0
	connections	0.33	0.63	-	0.03
BS-HA-NEAT	accuracy	1	1	1	-
	generations	1	1	1	-
	hidden nodes	0.26	0.71	0	-
	connections	1	1	0.03	-

that the extended algorithms BS-NEAT and BS-HA-NEAT are able to solve problems that were not solvable before. Although the inspiration for this behavioral speciation mechanism came from HA-NEAT, its successful application also on NEAT shows the potential to other NEAT-based algorithms as well.

In future work we are going to extend these experiments by testing behavioral speciation on other extensions of NEAT e.g. on FD-NEAT [9] and investigate the influence of the k-means algorithm used for clustering the individuals into species. Finally, we are going to test the proposed behavioral speciation mechanism on a real world problem of lung nodule classification as the one presented in [15].

TABLE IX: P value of Kruskal Wallis hypothesis tests with Bonferonni Correction for the Spiral Plots Dataset

	Metric	NEAT	BS-NEAT	HA-NEAT	BS-HA-NEAT
NEAT	accuracy	-	0	1	0.82
	generations	-	1	1	1
	hidden nodes	-	0	0.07	0
	connections	-	0	0.15	0
BS-NEAT	accuracy	0	-	0	0
	generations	1	-	1	1
	hidden nodes	0	-	0.03	0.8
	connections	0	-	0.07	0.59
HA-NEAT	accuracy	1	0	-	0.04
	generations	1	1	-	1
	hidden nodes	0.07	0.03	-	0
	connections	0.15	0.07	-	0
BS-HA-NEAT	accuracy	0.82	0	0.04	-
	generations	1	1	1	-
	hidden nodes	0	0.8	0	-
	connections	0	0.59	0	-

REFERENCES

[1] J. Lehman and R. Miikkulainen, "Neuroevolution," *Scholarpedia*, vol. 8, no. 6, p. 30977, 2013.

[2] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary Computation*, vol. 10, no. 2, pp. 99–127, 2002.

[3] A. Hagg, M. Mensing, and A. Asteroth, "Evolving parsimonious networks by mixing activation functions," in *Proceedings of the Genetic and Evolutionary Computation Conference*, ser. GECCO '17. New York, NY, USA: ACM, 2017, pp. 425–432. [Online]. Available: http://doi.acm.org/10.1145/3071178.3071275

[4] K. O. Stanley, "Compositional pattern producing networks: A novel abstraction of development," *Genetic programming and evolvable machines*, vol. 8, no. 2, pp. 131–162, 2007.

[5] K. O. Stanley, D. B. D'Ambrosio, and J. Gauci, "A hypercube-based encoding for evolving large-scale neural networks," *Artificial life*, vol. 15, no. 2, pp. 185–212, 2009.

[6] L. Trujillo, G. Olague, E. Lutton, F. F. De Vega, L. Dozal, and E. Clemente, "Speciation in behavioral space for evolutionary robotics," *Journal of Intelligent & Robotic Systems*, vol. 64, no. 3-4, pp. 323–351, 2011.

[7] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[8] E. Papavasileiou and B. Jansen, "The importance of the activation function in neuroevolution with fs-neat and fd-neat," in *Computational Intelligence (SSCI), 2017 IEEE Symposium Series on*. IEEE, 2017, pp. 1–7.

[9] M. Tan, M. Hartley, M. Bister, and R. Deklerck, "Automated feature selection in neuroevolution," *Evolutionary Intelligence*, vol. 1, no. 4, pp. 271–292, 2009.

[10] L. Trujillo, G. Olague, E. Lutton, and F. F. de Vega, "Behavior-based speciation for evolutionary robotics," in *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, 2008, pp. 297–298.

[11] J. Huizinga, J. Clune, and J.-B. Mouret, "Evolving neural networks that are both modular and regular: Hyperneat plus the connection cost technique," in *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, 2014, pp. 697–704.

[12] L. A. Belanche and F. F. González, "Review and evaluation of feature selection algorithms in synthetic problems," *arXiv preprint arXiv:1101.2320*, 2011.

[13] D. James and P. Tucker, "A comparative analysis of simplification and complexification in the evolution of neural network topologies," in *Proc. of Genetic and Evolutionary Computation Conference*, 2004.

[14] A. Hagg, "haneat," https://github.com/alexander-hagg/haneat-gecco2017, 2017.

[15] A. A. A. Setio, A. Traverso, T. De Bel, M. S. Berens, C. van den Bogaard, P. Cerello, H. Chen, Q. Dou, M. E. Fantacci, B. Geurts *et al.*, "Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: the luna16 challenge," *Medical image analysis*, vol. 42, pp. 1–13, 2017.